A New Method and Toolbox for Easily Calibrating Omnidirectional Cameras

Davide Scaramuzza[1] and Roland Siegwart[1]

[1] Swiss Federal Institute of Technology Zurich (ETHZ)
Autonomous Systems Lab, CLA-E, Tannenstrasse 3,
8092 Zurich, Switzerland
{davide.scaramuzza, r.siegwart}@ieee.org

Abstract. In this paper, we focus on calibration of central omnidirectional cameras, both dioptric and catadioptric. We describe our novel camera model and algorithm and provide a practical Matlab Toolbox, which implements the proposed method. Our method relies on the use of a planar grid that is shown by the user at different unknown positions and orientations. The user is only asked to click on the corner points of the images of this grid. Then, calibration is quickly and automatically performed. In contrast with previous approaches, we do not use any specific model of the omnidirectional sensor. Conversely, we assume that the imaging function can be described by a polynomial approximation whose coefficients are estimated by solving a linear least squares minimization problem followed by a non-linear refinement. The performance of the approach is shown through several calibration experiments on both simulated and real data. The proposed algorithm is implemented as a Matlab Toolbox, which allows any inexpert user to easily calibrate his own camera. The toolbox is completely Open Source and is freely downloadable from the author's Web page.

Keywords: catadioptric, dioptric, fish eye, omnidirectional, camera, calibration, Matlab, toolbox, ocamcalib.

1 Introduction

An omnidirectional camera is a vision system providing a 360° field of 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 by employing purely dioptric fisheye lenses. Furthermore, 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 fisheye lenses cannot in general be considered as central systems, but the single viewpoint property holds approximately true for some camera models [2].

Proceedings of the ICVS Workshop on Camera Calibration Methods for Computer Vision Systems - CCMVS2007
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

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).

In the last years, novel calibration techniques have been developed, which apply to any central omnidirectional camera. For instance, in [2], the authors extend their self-calibration procedure described in [8] to mirrors, fisheye lenses, and non-central cameras. In [12] and [13], 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.

The work described in this paper also handles with calibration of any central omnidirectional camera but aims at providing a technique that requires the minimum user interaction and that is very easy to apply also for the inexpert user. Indeed, our technique requires the use of a chessboard-like pattern that is shown by the user at a few different unknown positions and orientations. Then, the user is only asked to click on the corner points of the images of the pattern. The strong point of our technique resides in the use of a new generalized camera model that adjusts according to the appearance of the pattern in the omnidirectional images.

The method described in this paper has been already introduced in [14] and [15]. The novelty of the method resides in the use of a generalized parametric model of the sensor, which is suitable to different kinds of omnidirectional vision systems, both catadioptric and dioptric. In this model, we assume that the imaging function, which manages the projection of a 3D real point onto a pixel point on the image, can be described by a polynomial approximation whose coefficients are the parameters to be calibrated. This imaging model encapsulates both the intrinsic parameters of the camera and the parameters of the mirror without using any specific model for them. We show that this model is able to accurately calibrate any central omnidirectional camera. Moreover, because of its polynomial form, its parameters turn out to be easy to compute. Indeed, the computation is performed by linear least-square minimization followed by a non-linear refinement based on the maximum likelihood criterion.

Another important aspect of our technique is the automatic detection of the center of the omnidirectional camera. Indeed, in previous works, the detection of the center is performed by exploiting the visibility of the circular external boundary of the mirror. In those works, the mirror boundary is first enhanced by using an edge detector, and then, a circle is fitted to the edge points to identify the location of the center. In our approach, we no longer need the visibility of the mirror boundary. The algorithm described in this paper is based on an iterative procedure that uses only the corner points selected by the user.

This paper is organized in the following way. In section 2, we will present our camera model. In section 3, we will describe our calibration technique and the automatic detection of the center of distortion. In section 5 and 6, we will show the performance of the approach through several calibration experiments on both simulated and real

data. The reader can find a complete implementation of the proposed algorithm in the form of a Matlab Toolbox named "OCamCalib". We will give an overview of the toolbox in section 4. The OCamCalib toolbox is completely Open Source and is freely downloadable from the author's Web page [16]. Because of its ease of use, the toolbox turns out to be very practical, and allows any inexpert user to calibrate his own omnidirectional camera.

2 Camera Model

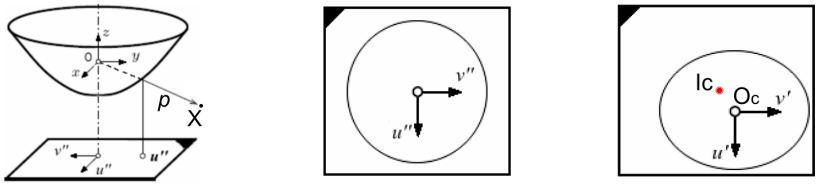
In this section, we describe our omnidirectional camera model. In the general central camera model, we identify two distinct reference systems: the camera image plane (u', v') and the sensor plane (u'', v''). The camera image plane coincides with the camera CCD, where the points are expressed in pixel coordinates. The sensor plane is a hypothetical plane orthogonal to the mirror axis, with the origin located at the plane-axis intersection.
In figure 1, the two reference planes are shown for 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. 1.a).
Let $\boldsymbol{X}$ be a scene point. Then, assume $\boldsymbol{u''} = [u'', v'']^T$ be the projection of $\boldsymbol{X}$ onto the sensor plane, and $\boldsymbol{u'} = [u', v']^T$ its image in the camera plane (Fig. 1.b and 1.c). 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 $\boldsymbol{g}$, which captures the relationship between a point $\boldsymbol{u''}$, in the sensor plane, and the vector $\boldsymbol{p}$ emanating from the viewpoint O to a scene point $\boldsymbol{X}$ (see figure 1.a). By doing so, the relation between a pixel point $\boldsymbol{u'}$ and a scene point $\boldsymbol{X}$ is:

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

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 $\boldsymbol{g}$, so that all vectors $g(\mathrm{A}\boldsymbol{u'} + \mathrm{t})$ satisfy the projection equation (1). We assume for $\boldsymbol{g}$ the following expression

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

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



(a) (b) (c)

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 depending on the mirror or the lens construction. These functions can be found in [10, 11, and 17]. Unlike using a specific model of the sensor, we chose 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 those sensors where the single viewpoint property is not exactly verified (e.g. generic fisheye cameras). As we proved in our earlier work [15], we can use the following polynomial form for f

$$f(u'',v'') = a_0 + a_2\rho''^2 + ... + a_N\rho''^N , \tag{3}$$

where the coefficients a_i, $i = 0, 2, ...\mathrm{N}$ and the polynomial degree N are the calibration parameters that we want to determine. Observe that the first order term has been removed [15]. To resume, equation (1) can be rewritten in this way:

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

3. Camera Calibration

3.1 Solving for Intrinsic and Extrinsic Parameters

According to what we told so far, to calibrate an omnidirectional camera, we have to estimate the parameters A, t, $a_0, a_2, ...$, and a_N.

In our approach, we decided to split the estimation of these parameters into two stages. In one, we estimate the affine parameters A and t. In the other one, we estimate the coefficients $a_0, a_2, ...$, and a_N.

The parameters A and t describe the affine transformation that relates the sensor plane to the camera plane (figures 1.b and 1.c). A is the stretch matrix and t is the translation vector $\overline{I_c O_c}$ (figure 1.c). To estimate A and t we introduce a method, which, unlike other previous works, does not require the visibility of the circular external boundary

of the mirror (sketched by the ellipse in figure 1.c). This method is based on an iterative procedure, which starts by setting A to the identity matrix *Eye* and $t=0$. This assumption means that the camera plane and the sensor plane initially coincide. The correct elements of A will be estimated afterwards by non-linear refinement, while t will be estimated by an iterative search algorithm. This approach will be detailed in section 3.2 and 3.3.

According to this, from now on we assume $A=Eye$ and $t=0$, which means $\boldsymbol{u''} = \boldsymbol{u'}$. Thus, by substituting this relation in (4) and using (3), we have the following projection equation

$$\lambda \cdot \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda \cdot \boldsymbol{g}(\boldsymbol{u'}) = \lambda \cdot \begin{bmatrix} u' \\ v' \\ f(\rho') \end{bmatrix} = \lambda \cdot \begin{bmatrix} u' \\ v' \\ a_0 + a_2\rho'^2 + \ldots + a_N\rho'^N \end{bmatrix} = \mathrm{P} \cdot \boldsymbol{X}, \quad \lambda > 0 \ , \qquad (5)$$

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, observe that now only N parameters ($a_0, a_2, \ldots, a_N$) need to be estimated. From now on, we will refer to these parameters as intrinsic parameters.

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 $R \in \Re^{3x3}$ and a translation $T \in \Re^{3x1}$. R and T will be referred to as 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 (5) becomes:

$$\lambda_{ij} \cdot \boldsymbol{p_{ij}} = \lambda_{ij} \cdot \begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + a_2\rho'^2 + \ldots + a_N\rho'^N \end{bmatrix} = \mathrm{P}^{\mathrm{i}} \cdot \boldsymbol{X} = \begin{bmatrix} \boldsymbol{r_1^i} & \boldsymbol{r_2^i} & \boldsymbol{r_3^i} & T^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r_1^i} & \boldsymbol{r_2^i} & T^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix}, \qquad (6)$$

where $\boldsymbol{r_1}, \boldsymbol{r_2}$ and $\boldsymbol{r_3}$ are the column vectors of R.

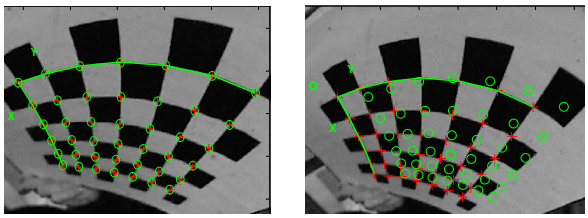
Therefore, in order to solve for camera calibration, the extrinsic parameters have also to be determined for each pose of the calibration pattern. Observing equation (6), we can eliminate the dependence from the depth scale λ_{ij} by multiplying both sides of the equation vectorially by $\boldsymbol{p_{ij}}$. This implies that each point p_j contributes three homogeneous non linear equations:

$$\begin{cases} v_j \cdot (r_{31}X_j + r_{32}Y_j + t_3) - f(\rho_j) \cdot (r_{21}X_j + r_{22}Y_j + t_2) = 0 & (7.1) \\ f(\rho_j) \cdot (r_{11}X_j + r_{12}Y_j + t_1) - u_j \cdot (r_{31}X_j + r_{32}Y_j + t_3) = 0 & (7.2) \\ u_j \cdot (r_{21}X_j + r_{22}Y_j + t_2) - v_j \cdot (r_{11}X_j + r_{12}Y_j + t_1) = 0 & (7.3) \end{cases}$$

where the sub-index *i* has been removed to lighten the notation, and t_1, t_2 and t_3 are the elements of *T*. Observe that in (7), X_j, Y_j and Z_j are known, and so are u_j, v_j. Also, observe that only (7.3) is linear in the unknown $r_{11}, r_{12}, r_{21}, r_{22}, t_1, t_2$. From now on, the details for the resolution of equation (7) can be found in [15]. The principle of the technique consists in first solving for the parameters $r_{11}, r_{12}, r_{21}, r_{22}, t_1$, and t_2 by linearly solving equation (7.3). Next, we use the solution of (7.3) as input to (7.1) and (7.2), and solve for the remaining parameters $a_0, a_2, ..., a_N$ and t_3. In both steps, the solution is achieved by using linear least-square minimization.

3.2 Detection of the Image Center

As stated in section 1, a peculiarity of our calibration toolbox is that it requires the minimum user interaction. One of the tools that accomplish this task is its capability of identifying the center of the omnidirectional image O_c (figure 1.c) even when the external boundary of the sensor is not visible in the image.
At the beginning of section 3.1, we made the following assumptions for *A* and *t*, namely *A=Eye* and *t=0*. Then, we derived the equations for solving for the intrinsic and extrinsic parameters that are valid only under those assumptions.



(a) (b)

Fig. 2. Corret position if the center (a). Wrong position of the center (b).

In figure 2.a, the reader can see what happens when the position of the center is correct. The red crosses are the input calibration points selected by the user. The green rounds are the 3D points reprojected onto the images according to the intrinsic and extrinsic parameters estimated by the calibration. As the reader can see, the 3D points perfectly overlay the input points, meaning that the calibration worked properly. Figure 2.b shows the result when the input position of the center is wrong, that is, the reprojection error is large. 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.
This result leads us to an exhaustive search of the center O_c, which stops when the difference between two potential center locations is smaller than a certain ε (we used ε=0.5 pixels). The algorithm is the following:

1. At each step of this iterative search, a fixed number of candidate center locations is uniformly selected from a given image region.

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 providing the minimum SSRE is taken as a potential center.
4. The search proceeds by selecting other candidate locations 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 less than 3 seconds to stop. At this point, the reader might be wondering how we do estimate the elements of matrix *A*. In fact, at the beginning we assumed *A=Eye*. The iterative algorithm mentioned above exhaustively searches the location of the center (namely O_c) by leaving *A* unchanged. The reason for doing so is that the eccentricity of the external boundary of an omnidirectional image is usually close to zero, which means *A~Eye*. Therefore, we decided to estimate *A* in a second stage by using a non linear minimization method, which is described in section 3.3.

3.3 Non-Linear Refinement

The linear solution given in section 3.1 is obtained through minimizing an algebraic distance, which is not physically meaningful. To this end, we chose to refine the calibration parameters through maximum likelihood inference. Let us assume that we are given *K* images of the calibration grid, 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:

$$E = \sum_{i=1}^{K} \sum_{j=1}^{L} \left\| m_{ij} - \hat{m}\left(R_i, \boldsymbol{T}_i, A, O_c, a_0, a_2, ..., a_N, \boldsymbol{M}_j\right) \right\|^2 , \qquad (8)$$

where $\hat{m}\left(R_i, \boldsymbol{T}_i, A, O_c, a_0, a_2, ..., a_N, \boldsymbol{M}_j\right)$ is the reprojection of the point $\boldsymbol{M}_j$ of the grid *i* according to equation (1). R_i and T_i are the rotation and translation matrices of each grid 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 stretch matrix *A* and the center of the omnidirectional image O_c. To minimize (8), we used the Levenberg-Marquadt algorithm as implemented in the Matlab function *lsqnonlin*. The algorithm requires an initial guess for the parameters. These initial parameters are the ones obtained using the linear technique described in section 3.1. As a first guess for *A*, we used the identity matrix, while for O_c we used the position estimated through the iterative procedure explained in section 3.2.

4. Introduction to the OCamCalib Toolbox for Matlab

The reason why we implemented the OCamCalib Toolbox for Matlab is to allow any user to easily and quickly calibrate his own omnidirectional camera. The OCamCalib toolbox can be freely downloaded from the Author's web page [16] (or google for "ocamcalib"). The outstanding features of the toolbox are the following:

- Capability of calibrating different kinds of central omnidirectional cameras without any knowledge about the parameters of the camera or about the shape of the mirror.
- Automatic detection of the center.
- Visual feedback about the quality of the calibration result by reprojecting the 3D points onto the input images.
- Computer assisted selection of the input points. Indeed, a corner detector assists the selection of the corner points on the calibration pattern.

5 Experimental Results on Simulated Data

5.1 Performance with respect to the noise level

We evaluated the performance of our algorithm through calibration experiments both on synthetic and real images. In particular, we used synthetic images to study the robustness of our calibration technique in case of inaccuracy in detecting the calibration points. To this end, we generated several synthetic poses of a calibration pattern. Then, Gaussian noise with zero mean and standard deviation σ was added to the projected image points. We varied the noise level from σ=0.1 to σ=3.0 pixels, and, for each noise level, we performed 100 independent calibration trials and computed the mean reprojection error (SSRE).

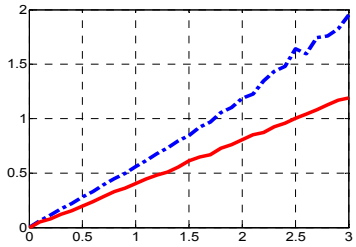


Fig. 3. Mean SSRE versus σ. When using the linear minimization alone (dashed line) or along with the non-linear refinement (solid line).

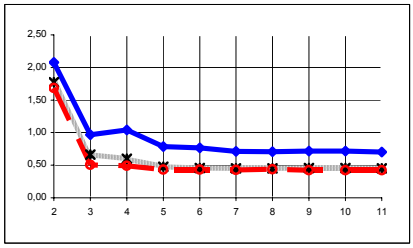


Fig. 4 Mean SSRE versus the number of images for different polynomial degrees: 2nd order (black ×), 3rd order (blue •) and 4th order (red ◦).

Figure 3 shows the plot of the mean SSRE as a function of σ. Observe that we separated the results obtained by using the linear minimization alone from the results of the non-linear refinement. As the reader can see, in both cases the average error increases linearly with the noise level. Furthermore, the reprojection error of the non-linear estimation keeps always smaller than the error computed by the linear method. Finally, notice that when σ=1.0, which is larger than the normal noise in practical situations, the SSRE of the non-linear method is lower than 0.4 pixels.
In Fig. 5, we show the 3D points of a simulated grid reprojected onto the image. The ground truth is represented by red crosses, while the blue rounds represent the calibration points perturbed by noise with σ=3.0 pixels. Notice that, despite the large

amount of noise, the calibration is able to compensate for the large error introduced. In fact, after calibration, the reprojected calibration points (red squares) are very close to the ground truth.
We also want to evaluate the accuracy in estimating the extrinsic parameters R and T of each calibration plane. To this end, figure 6 shows the plots of the absolute error (measured in *mm*) in estimating the origin coordinates (*x*, *y* and *z*) of a given grid. The absolute error is very small because it keeps always smaller than 2*mm*. Even though we do not show the plots here, we also evaluated the error in estimating the correct plane orientations, and we got an average absolute error less than 2°.

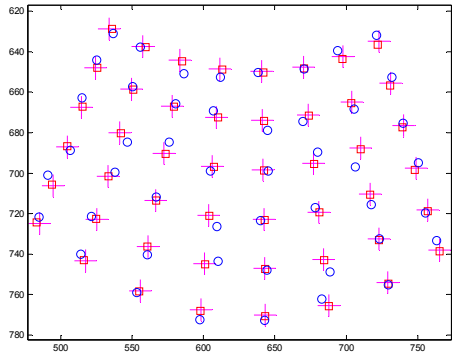


Fig. 5. An image of a simulated calibration grid, projected onto the simulated omnidirectional image. Calibration points are affected by noise with σ =3.0 pixels (blue rounds). Ground truth (red crosses). Reprojected points after the calibration (red squares).

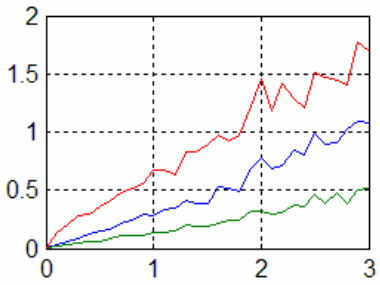


Fig. 6. Accuracy of the extrinsic parameters: the absolute error (*mm*) of the translation vector vs. de noise level (*pixels*). The error along the x, y and z coordinates is represented respectively in red, blue and green.

6 Experimental Results on Real Data

6.1 Performance w.r.t. the Number of Planes and to the Polynomial degree

In this section, we present the calibration performance on real data. In these experiments, we calibrated a catadioptric system made up of a KAIDAN 360° One VR hyperbolic mirror and a SONY CCD camera the resolution of 900x1200 pixels. In the first experiment, we investigated the performance of our technique with respect to the number of images of the calibration grid, for a given polynomial degree. We varied the number of pictures from 2 to 11, and for each set we performed the calibration and computed the SSRE. The SSRE versus the number of images is plotted in Fig. 4 for different polynomial degrees. Note that the error decreases as the number of images augments. Moreover, by using a fourth order polynomial, we obtain the minimum SSRE value. A third order polynomial also provides a similar performance when more than four images are taken. Conversely, when using a second

order polynomial, the SSRE keeps higher. Thus, in practical applications we always use a fourth order approximation. In calibrating our omnidirectional camera, we got a reprojection error of 0.3 pixels using a fourth order polynomial.

6.2 Structure from Motion

An indirect method to evaluate the quality of the calibration of a real camera consists in reconstructing the 3D structure of an object from its images and checking then the quality of the reconstruction. This problem is well known by the computer vision community as structure from motion. The object we used in this experiment is a trihedron made up of three orthogonal chessboard-like patterns of known geometry (Fig. 7). After calibrating the camera, we took two images of the trihedron from two different unknown positions (Fig. 7). Next, we picked up several point matches from both views and we applied the Eight-Point algorithm [18]. The results of the reconstruction, after rendering, are shown in Fig. 7. As the reconstruction with one single camera can be done up to a scale factor, we recovered the scale factor by comparing the average size of a reconstructed checker with the real size on the trihedron. In the end, we computed the angles between the three planes fitting the reconstructed points and we found the following values: 94.6°, 86.8° and 85.3°. Moreover, the average distances of these points from the fitted planes were respectively 0.05 cm, 0.75 cm and 0.07 cm. Finally, being the size of each checker 6.0 cm x 6.0 cm, we also calculated the dimension of every reconstructed checker and we found an average error of 0.3 cm. These results comply with the expected orthogonality of the surfaces and the size of the checkers in the ground truth.

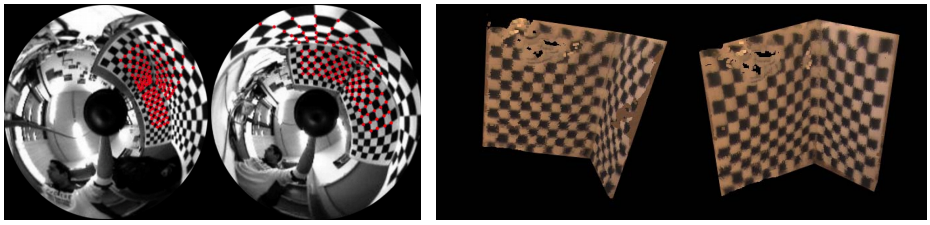


Fig. 7. The input images for the 8-Point algorithm (left) and the reconstruction results (right).

6.3 Mapping Color Information on 3D point clouds from a Laser Range Finder

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 laser range finder (SICK LMS200 [19]). 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. 8. In order to perform this mapping both the intrinsic and extrinsic parameters have to be accurately determined. Now, the extrinsic parameters describe position and orientation of the camera frame with respect to the laser reference frame. Observe that even small errors in estimating the correct intrinsic and extrinsic parameters would produce a large offset into the output colorful map. In this

experiment, the colors well reprojected onto the 3D structure of the environment, showing that the calibration was accurately done.

Fig. 8. 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 calibration, the color information was mapped onto the 3D points extracted by a rotating sick laser range finder. In the lower windows are the results of the mapping. The colors are well reprojected onto the 3D structure of the environment, showing that the camera calibration was accurately done.

7 Conclusions

In this paper, we presented a method for calibrating central omnidirectional cameras both dioptric or catadioptric. The method relies on a generalized parametric function that describes the relation between a given pixel point and the correspondent 3D vector emanating from the single effective viewpoint of the camera. We describe this function by means of a polynomial approximation whose coefficients are the parameters to be calibrated. The performance of the approach was evaluated through several experiments both on synthetic and real data. The results showed that this technique is very accurate as the mean reprojection error is smaller than 0.4 pixels in practical situations. We also showed the accuracy of the result by performing structure from motion with a calibrated camera. The results showed a very good agreement between the ground truth and the reconstructed object. Another performance evaluation was for projecting the color information of an image onto real 3D points coming from a 3D sick laser range finder. To make our method usable and applicable for any user, we implemented the whole algorithm in the form of a Matlab Toolbox named OCamCalib [16]. The toolbox is completely Open Source and can be freely downloaded from the author's Web page [16]. Because of its ease of use, the OCamCalib toolbox turns out to be very practical, and allows any inexpert user to calibrate his own omnidirectional camera.

References

1. Baker, S. & Nayar, S.K. A theory of catadioptric image formation. Proceedings of the 6th International Conference on Computer Vision, pp. 35–42, ISBN 81-7319-221-9, India, January 1998, IEEE Computer Society, Bombay (1998).
2. Micusik, B. & Pajdla, T. Autocalibration & 3D reconstruction with non-central catadioptric cameras. Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 58-65, ISBN 0-7695-2158-4, US, June 2004, IEEE Computer Society, Washington, (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. Kumler, J. & Bauer, M. Fisheye lens designs and their relative performance. Proceedings of SPIE Conference. Vol. 4093. pp. 360-369, (2000).
11. B.Micusik, D.Martinec, T.Pajdla. 3D Metric Reconstruction from Uncalibrated Omnidirectional Images. ACCV 2004, Korea January (2004).
12. Ying, X. & Hu, Z. Catadioptric Camera Calibration Using Geometric Invariants, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 10, October 2004, pp. 1260-1271, ISSN: 0162-8828, (2004).
13. Barreto, J. & Araujo, H. Geometric Properties of Central Catadioptric Line Images and their Application in Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp. 1327-1333, August, (2005).
14. Scaramuzza, D., Martinelli, A. & Siegwart, R. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion, Proceedings of IEEE International Conf. on Computer Vision Systems, USA, New York, January (2006).
15. Scaramuzza, D., Martinelli, A. & Siegwart, R. A Toolbox for Easy calibrating Omnidirectional Cameras. Proceedings of the IEEE International Conference on Intelligent Robots and Systems, pp. 5695–5701, China, Beijing, October, (2006).
16. To download the toolbox google for "ocamcalib" or go the author's Web page: http://asl.epfl.ch/~scaramuz/research/Davide_Scaramuzza_files/Research/OcamCalib_Tutorial.htm. (2007).
17. T. Svoboda, T.Pajdla. Epipolar Geometry for Central Catadioptric Cameras. IJCV, 49(1), pp. 23-37, Kluwer August (2002).
18. H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. Nature, 293:133–135, September, (1981).
19. Weingarten, J. and Siegwart, R. EKF-based 3D SLAM for Structured Environment Reconstruction. In Proceedings of IROS 2005, Edmonton, Canada, August, (2005).