17

A Practical Toolbox for Calibrating Omnidirectional Cameras

Davide Scaramuzza and Roland Siegwart
Swiss Federal Institute of Technology, Zurich
Switzerland

1. 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. Omnidirectional cameras can be classified into two classes, central and non-central, depending on whether they satisfy the single effective viewpoint property or not (Baker & Nayar, 1998). As noted in (Svoboda & T. Pajdla, 1997), it is highly desirable that such imaging systems have a single effective viewpoint. When this property is verified, there exists a single center of projection, that is, every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. The reason a single viewpoint is so desirable is that it allows the user to generate geometrically correct perspective images from the pictures captured by an omnidirectional camera. Moreover, it allows applying the known theory of epipolar geometry, which easily allows the user to perform ego-motion estimation and structure from motion from image correspondences only.

As shown in (Baker & Nayar, 1998), 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 central systems, but the single viewpoint property holds approximately true for some camera models (Micusik & Pajdla, 2003).

In this chapter, we focus on calibration of central omnidirectional cameras, both dioptric and catadioptric. After outlining previous works on omnidirectional camera calibration, we describe our novel procedure and provide a practical Matlab Toolbox, which allows any inexpert user to easily calibrate his own camera.

Accurate calibration of a vision system is necessary for any computer vision task requiring extracting metric information of the environment from 2D images, like in ego-motion estimation and structure from motion. While a number of calibration methods has been developed for standard perspective cameras (Zhang, 2000), little work on omnidirectional cameras has been done. The first part of this chapter will present a short overview about previous methods for calibration of omnidirectional cameras. In particular, their limitations will be pointed out. The second part of this chapter will present our calibration technique whose performance is evaluated through calibration experiments. Then, we will present our

Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

Matlab toolbox (that is freely available on-line), which implements the proposed calibration procedure. We will also describe features and use of our toolbox.

2. 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 (Cauchois et al., 2000; Bakstein & Pajdla, 2002) or plumb lines (Geyer & Daniilidis, 2002). The second group covers techniques that do not use this knowledge. The latter includes calibration methods from pure rotation or planar motion of the camera (Gluckman & Nayar, 1998), and self-calibration procedures, which are performed from point correspondences and epipolar constraint through minimizing an objective function (Kang, 2000; Micusik & Pajdla, 2003).

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 expensive equipment (Bakstein & Pajdla, 2002; Gluckman & Nayar, 1998). For instance, in (Bakstein & Pajdla, 2002), a fish-eye lens with a 183° field of view is used as an omnidirectional sensor. Then, the calibration is performed by using a half-cylindrical calibration pattern perpendicular to the camera sensor, which rotates on a turntable.

In (Geyer & Daniilidis, 2002; Kang, 2000), the authors treat the case of a parabolic mirror. In (Geyer & Daniilidis, 2002), it is shown that vanishing points lie on a conic section which encodes the entire calibration information. Thus, the projections of two sets of parallel lines suffice for the intrinsic camera calibration. However, this property does not apply to non-parabolic mirrors. Therefore, the proposed technique cannot be easily generalized to other kinds of sensors.

In contrast with the techniques mentioned so far, the methods described in (Kang, 2000; Micusik & Pajdla, 2003; Micusik et al., 2004) fall in the self-calibration category. These methods require no calibration pattern, nor a priori knowledge about the scene. The only assumption is the capability to automatically find point correspondences in a set of panoramic images of the same scene. Then, calibration is directly performed by epipolar geometry by minimizing an objective function. In (Kang, 2000), this is done by employing a parabolic mirror, while in (Micusik & Pajdla, 2003; Micusik et al., 2004) a fish-eye lens with a view angle greater than 180° is used. However, besides focusing on particular sensor types, the mentioned self-calibration techniques may suffer in case of tracking difficulties and of a small number of features points (Bougnoux, 1998).

The calibration methods described so far focus on particular sensor types, such as parabolic and hyperbolic mirrors or fish-eye lenses. In contrast with these methods, in the last years, novel calibration techniques have been developed, which apply to any central omnidirectional camera. For instance, in (Micusik & Pajdla, 2004), the authors extend the geometric distortion model and the self-calibration procedure described in (Micusik & Pajdla, 2003), including mirrors, fish-eye lenses, and non-central cameras. In (Ying & Hu, 2004; Barreto & Araujo, 2005), 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 chapter also handles with calibration of any central omnidirectional camera but aims at providing a technique 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 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 camera model that adapts according to the appearance of the pattern in the omnidirectional images. The peculiarity of this model is that it can also handle the cases where the single effective viewpoint property is not perfectly satisfied. Indeed, although several omnidirectional cameras exist directly manufactured to have this property, for a catadioptric system this requires to accurately align the camera and the mirror axes. In addition, the focus point of the mirror has to coincide with the optical center of the camera. Since it is very difficult to avoid camera-mirror misalignments, an incorrectly aligned catadioptric sensor can lead to a quasi single viewpoint system (Swaminathan & Grossberg, 2001).

The method described in this chapter was first introduced in (Scaramuzza et al., 2006). In that work, we proposed a generalized parametric model of the sensor, which is suitable to different kinds of omnidirectional vision systems, both catadioptric and dioptric. In that model, we assume that the imaging function, which manages the projection of a 3D real point onto a pixel of the image plane, can be described by a Taylor series expansion whose coefficients are the parameters to be calibrated.

In this chapter, we will first summarize the generalized camera model (section 3) and the calibration method introduced in our previous work (section 4). Then, in section 5, we will introduce our Matlab Toolbox (named OcamCalib Toolbox). There, we will outline the features of the toolbox, with particular regard to 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 chapter is based on an iterative procedure that uses only the points selected by the user.

In section 6, the performance of our toolbox will be evaluated through calibration experiments.

3. Omnidirectional 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 Figure 1.a).

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 (Figure 1.b and 1.c). As observed in (Micusik & Pajdla, 2003), 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 $\mathrm{t} \in \Re^{2x1}$.

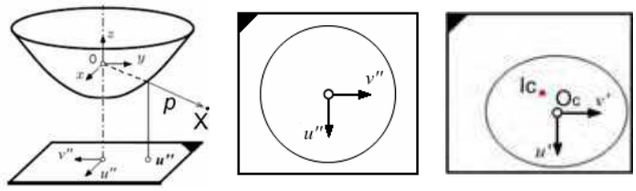
At this point, we can introduce the imaging function $\boldsymbol{g}$, which captures the relationship between a point $\boldsymbol{u''}$, in the sensor plane, and the vector $\boldsymbol{p}$ emanating from the viewpoint O to a scene point $\boldsymbol{X}$ (see figure 1.a). By doing so, the relation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

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

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)

Figure 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 (Kumler & Bauer, 2000), (Micusik et al., 2004), and (Svoboda & Pajdla, 2002). 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 those sensors where the single viewpoint property is not exactly verified (e.g. generic fisheye cameras). We propose the following polynomial form for f

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

where the coefficients a_i, $i = 0, 1, 2, \ldots \mathrm{N}$ and the polynomial degree N are the calibration parameters that we want to determine. This polynomial description of f can be more simplified by considering that all previous definitions of f always satisfy the following:

$$\left.\frac{df}{d\rho}\right|_{\rho=0} = 0 \tag{4}$$

This property holds for hyperbolic and parabolic mirrors or fisheye cameras (see (Kumler & Bauer, 2000), (Micusik et al., 2004), and (Svoboda & Pajdla, 2002)).
This simplification allows us to assume $a_1 = 0$, and thus (3) can be rewritten as:

$$f(u'',v'') = a_0 + a_2\rho''^2 + ... + a_N\rho''^N \tag{5}$$

As a consequence, we reduced the number of parameters to be estimated.
To resume, equation (1) can be rewritten as

$$\lambda \cdot \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda \cdot \boldsymbol{g}(\mathrm{A}\boldsymbol{u}'+\mathrm{t}) = \lambda \cdot \begin{bmatrix} (\mathrm{A}\boldsymbol{u}'+\mathrm{t}) \\ f(u'',v'') \end{bmatrix} = \mathrm{P}\cdot\boldsymbol{X}, \qquad \lambda > 0 \tag{6}$$

4. Camera Calibration

4.1 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 separate 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 4.3.
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 (6) and using (5), we have the following projection equation

$$\lambda \cdot \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda \cdot \boldsymbol{g}(\boldsymbol{u}') = \lambda \cdot \begin{bmatrix} u' \\ v' \\ f(\rho') \end{bmatrix} = \lambda \cdot \begin{bmatrix} u' \\ v' \\ a_0 + a_2\rho'^2 + ... + a_N\rho'^N \end{bmatrix} = \mathrm{P}\cdot\boldsymbol{X}, \ \lambda > 0 \tag{7}$$

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, observe that now only N parameters ($a_0, a_2, ..., a_N$) 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 (7) becomes:

$$\lambda_{ij} \cdot \boldsymbol{p}_{ij} = \lambda_{ij} \cdot \begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + a_2\rho'^2 + \ldots + a_N\rho'^N \end{bmatrix} = \mathrm{P}^{\mathrm{i}} \cdot \boldsymbol{X} = \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & \boldsymbol{r}_3^i & T^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & T^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} \quad (8)$$

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 (8), 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

$$\begin{cases} v_j \cdot (r_{31}X_j + r_{32}Y_j + t_3) - f(\rho_j) \cdot (r_{21}X_j + r_{22}Y_j + t_2) = 0 & (9.1) \\ f(\rho_j) \cdot (r_{11}X_j + r_{12}Y_j + t_1) - u_j \cdot (r_{31}X_j + r_{32}Y_j + t_3) = 0 & (9.2) \\ u_j \cdot (r_{21}X_j + r_{22}Y_j + t_2) - v_j \cdot (r_{11}X_j + r_{12}Y_j + t_1) = 0 & (9.3) \end{cases}$$

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 (9), X_j, Y_j and Z_j are known, and so are u_j, v_j . Also, observe that only (9.3) 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 (9) can be found in (Scaramuzza et al., 2006). The principle of the technique consists first in solving for the parameters $r_{11}, r_{12}, r_{21}, r_{22}, t_1$, and t_2 by linearly solving equation (9.3). Next, we use the solution of (9.3) as input to (9.1) and (9.2), and solve for the remaining parameters $a_0, a_2, \ldots, a_N$ and t_3 . In both steps, the solution is achieved by using linear least-square minimization.

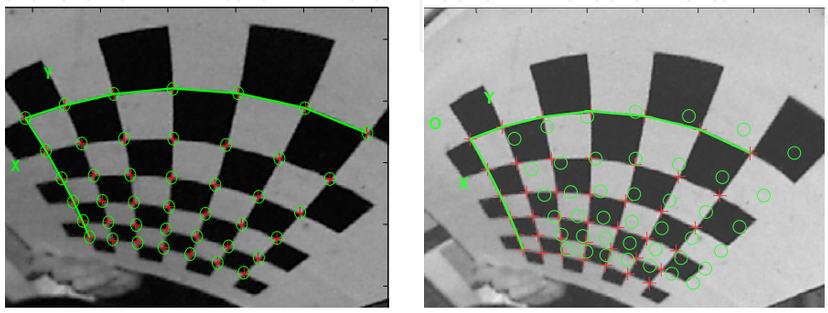
Up to now, we didn't specify which polynomial degree N one should use. To compute the best N, we actually start from N=2. Then, we increase N by unitary steps and we compute the average value of the reprojection error of all calibration points. The procedure stops when a minimum error is found. Typical empirical values for N are usually N=3 or N=4.

4.2 Detection of the Image Center

As stated in sections 1 and 2, 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 4.1, 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)

Figure 2. When the position of the center is correct, the 3D points of the checker board do correctly project (green rounds) onto the calibration points (red crosses) (a). Conversely, when the position of the center is wrong, the points do not project onto the real calibration points (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 (see figure 3).
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 chose to estimate A in a second stage by using a non linear minimization method, which is described in section 4.3.

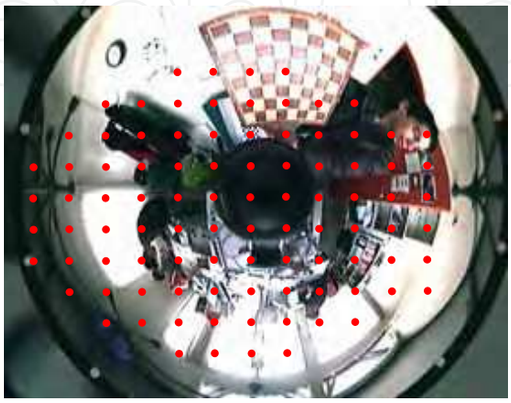


Figure 3. An omnidirectional image used for calibration with a chessboard used as a calibration pattern. The red points identify the candidate center locations taken during the first step of the algorithm. At each step, the candidate points occupy a smaller and smaller region around the final convergence point

4.3 Non Linear Refinement

The linear solution given in section 4.1 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 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:

$$E = \sum_{i=1}^{K} \sum_{j=1}^{L} \left\| m_{ij} - \hat{m}\left(R_i, \boldsymbol{T}_i, A, O_c, a_0, a_2, ..., a_N, \boldsymbol{M}_j\right) \right\|^2 \tag{10}$$

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 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 stretch matrix A and the center of the omnidirectional image O_c.

By minimizing the functional defined in (10), we actually find the calibration 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 (10), we used the Levenberg-Marquadt algorithm (Levenberg, 1944; Marquardt, 1963), 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 4.1. 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 subsection 4.2.

5. Introduction to the OcamCalib Toolbox for Matlab

The reason 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 Internet (e.g. 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, the selection of the corner points on the calibration pattern is assisted by a corner detector.

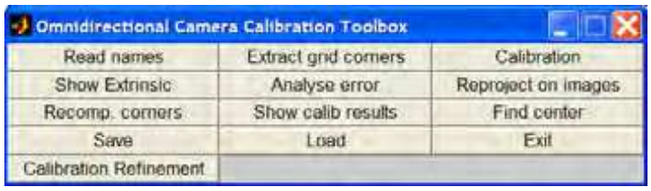


Figure 4. The graphical user interface of the OcamCalib Toolbox

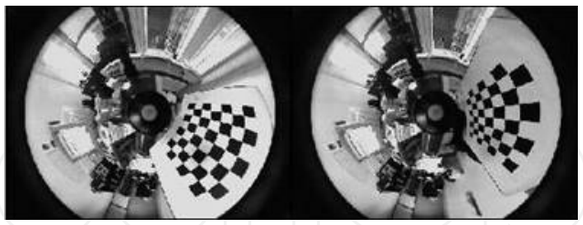


Figure 5. Some pictures with the checker board used as a calibration grid. In our experiments, we used at least 5 or more images with the grid shown all around the camera

The user interface of the toolbox is depicted in figure 4. After having collected a few pictures of a chessboard shown all around the omnidirectional camera (see figure 5), the images can be loaded for calibration (i.e. use "Read names"). In the second step, the user can start selecting the corner points of the pattern using the "Extracting grid corners" tool. By this tool, the user is asked to click on all the corner points by following the left-right order. To

achieve high accuracy in the selection of the input points, the clicking is assisted by a Harris base corner detector (Harris & Stephens, 1988).

In the third step, the calibration can be done by means of two tools. The "Calibration" tool will ask the user to specify the position of the center in case he knows, if not, the user can directly use the "Find center" tool, which automatically applies the iterative search algorithm described in 4.2. In both cases, the calibration is performed by using the linear estimation technique mentioned in 4.1. The optimal calibration parameters in the maximum likelihood sense can be estimated by the "Calibration Refinement" tool, which implements the non linear minimization described in 4.3. After the previous steps, the user can choose among several tools:

- "Show Extrinsic" visualizes the reconstructed 3D poses of the grid in the camera reference frame (figure 6).
- "Analyze error" visualizes the reprojection error of each calibration point along the x-y-axes.
- "Reproject on images" reprojects all the 3D points onto the images according to the calibrated parameters.
- "Recompute corners" attempts to automatically recompute the position of every corner point hand selected by the user. This is done by using the reprojected 3D points as initial guess locations for the corners.

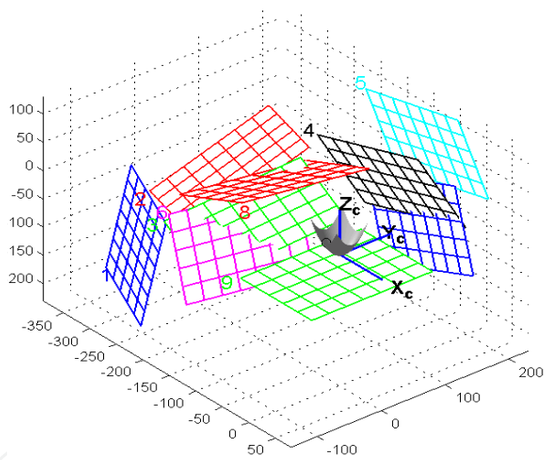


Figure 6. A picture of our simulator showing several calibration patterns and the virtual omnidirectional camera at the axis origin

After the calibration, all the parameter can be accessed through the structure "ocam_model". The calibrated camera model can then be used for other applications by means of the following two functions:

- **m = world2cam(M, ocam_model),** which reprojects a 3D point (`M`) onto the image and returns its pixel coordinates (`m`).
- **M = cam2world(m, ocam_model),** which, for every image point `m`, returns the 3D coordinates of the correspondent vector (`M`) emanating from the single effective viewpoint. This function is the inverse of the previous one.

6. Results

We evaluated the performance of our toolbox 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. Figure 7 shows the plot of the reprojection error 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 average reprojection error of the non linear method is lower than 0.4 pixels.

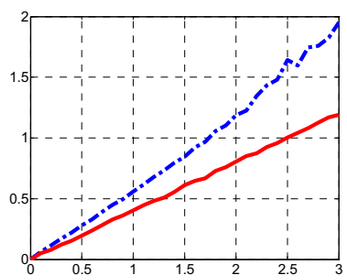


Figure 7. The reprojection error versus σ. The dashed line represents the results obtained by using the linear minimization alone. The solid line shows the results after the non linear refinement. Both units are considered in pixels

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 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 (see figure 8.a). Our omnidirectional camera is KAIDAN 360° One VR with a hyperbolic mirror.

After having calibrated the camera, we took two images of the trihedron from two different unknown positions (see figure 8.b). Next, several point matches were hand selected from both views of the object and the Eight Point algorithm was applied (Longuet-Higgins, 1981). In order to obtain good reconstruction results, more than eight points (we used 135 points) were used. The method mentioned so far gives a first good 3D reconstruction of the points. A better estimation of the 3D structure can be obtained by densely using all the pixels of the images. To accomplish this task, we used the first estimation along with normalized cross correlation to automatically match all the points of the image pair. Finally, all matches were used to compute the structure. The results of the reconstruction are shown in figure 8.c.

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.

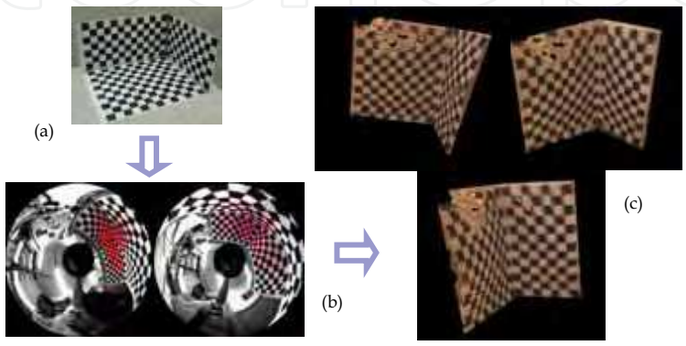


Figure 8. (a) The object to be reconstructed. (b) Two omnidirectional pictures of the object taken from two unknown positions. (c) Dense 3D reconstruction of the object. The reconstruction is very good, meaning that the model of the camera was well estimated

7. Conclusion

In this chapter, we presented a method for calibrating any central omnidirectional camera 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 view point of the camera. We describe this function by means of a polynomial expansion whose coefficients are the parameters to be calibrated.
Furthermore, we presented a toolbox for Matlab (named OcamCalib) that implements the mentioned calibration procedure. The toolbox is available on-line. We described the tools and the main features of our toolbox, one of which being the capability to automatically identify the center of the omnidirectional image. The toolbox relies on the use of a chessboard-like calibration pattern that is shown by the user at a few different positions and orientations. Then, the user is only asked to click on the corner points of the patterns. The performance of the toolbox was finally evaluated through experiments both on synthetic and real images. 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.

8. Acknowledgements

This work was conducted within the EU Integrated Projects COGNIRON (“The Cognitive Robot Companion”) and BACS (“Bayesian Approach to Cognitive Systems”). It was funded by the European Commission Division FP6-IST Future and Emerging Technologies under the contracts FP6-IST-002020 and FP6-IST-027140 respectively.

We want to thank Zoran Zivkovic and Olaf Booij, from the Intelligent Systems Laboratory of Amsterdam (University of Amsterdam), for providing the sample images included in our toolbox.
Furthermore, we want to thank Dr. Jean-Yves Bouguet, from Intel Corporation, for providing some functions used by our toolbox.

9. References

Baker, S. & Nayar, S.K. (1998). 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.

Svoboda, T., Pajdla T. & Hlavac, V. (1997). Central panoramic cameras: Geometry and design. *Research report K335/97/147*, Czech Technical University, Faculty of Electrical Engineering, Center for Machine Perception, Czech Republic, December 1997. Praha.

Micusik, B. & Pajdla, T. (2003). Estimation of omnidirectional camera model from epipolar geometry. *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. ISBN 0-7695-1900-8, US, June 2003, IEEE Computer Society, Madison.

Zhang, Z (2000). A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume 22, No. 11, November 2000, ISSN 0162-8828.

Kumler, J. & Bauer, M. (2000). Fisheye lens designs and their relative performance. *Proceedings of SPIE Conference.* Vol. 4093. pp. 360-369. 2000.

Micusik, B., Martinec, D. & Pajdla, T. (2004). 3D Metric Reconstruction from Uncalibrated Omnidirectional Images. *Proceedings of the Asian Conference on Computer Vision*. January 2004, Korea.

Svoboda, T. & Pajdla, T. (2001). Epipolar Geometry for Central Catadioptric Cameras. In. *Panoramic Vision: Sensors, Theory and Applications*, Benosman R. & Kang, S.B., pp. 85-114, Springer.

Cauchois, C., Brassart, E., Delahoche, L. & Delhommelle, T. (2000). Reconstruction with the calibrated SYCLOP sensor. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1493–1498, ISBN: 0-7803-6348-5, Japan, October 2000, IEEE Computer Society, Takamatsu.

Bakstein, H. & Pajdla, T. (2002). Panoramic mosaicing with a 180◦ field of view lens. *Proceedings of the IEEE Workshop on Omnidirectional Vision*, pp. 60–67, ISBN: 0-7695-1629-7, Denmark, June 2002, IEEE Computer Society, Copenhagen.

Geyer, C. & Daniilidis, K. (2002). Paracatadioptric camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May 2002, pp. 687-695, ISSN 0162-8828.

Gluckman, J. & Nayar, S. K. (1998). Ego-motion and omnidirectional cameras. *Proceedings of the 6th International Conference on Computer Vision*, pp. 999-1005, ISBN 81-7319-221-9, India, January 1998, IEEE Computer Society, Bombay.

Kang. S.B. (2000). Catadioptric self-calibration. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 201-207, ISBN: 0-7695-0662-3, USA, June 2000, IEEE Computer Society, Hilton Head Island.

Micusik, B., & Pajdla, T. (2004). Para-catadioptric Camera Auto-calibration from Epipolar Geometry. *Proceedings of the Asian Conference on Computer Vision*. January 2004, Korea.

Micusik, B., D.Martinec & Pajdla, T. (2004). 3D Metric Reconstruction from Uncalibrated Omnidirectional Images. *Proceedings of the Asian Conference on Computer Vision*. January 2004, Korea.

Bougnoux, S. (1998). From projective to Euclidean space under any practical situation, a criticism of self-calibration, *Proceedings of the 6th International Conference on Computer Vision*, pp. 790–796, ISBN 81-7319-221-9, India, January 1998, IEEE Computer Society, Bombay.

Swaminathan, R., Grossberg, M.D. & Nayar. S. K. (2001). Caustics of catadioptric cameras". *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2-9, ISBN: 0-7695-1143-0, Canada, July 2001, IEEE Computer Society, Vancouver.

Ying, X. & Hu, Z. (2004). 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.

Ying, X. & Hu, Z. (2004). Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model?, *Proceedings of the European Conference on Computer Vision*, pp. 442-455, Czech Republic, Lecture Notes in Computer Science, May 2004, Prague,

Barreto, J. & Araujo, H. (2005). 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.

Sturm, P. & Ramaligam, S. (2004). A Generic Concept for Camera Calibration, *Proceedings of the European Conference on Computer Vision*, pp. 1-13, Czech Republic, Lecture Notes in Computer Science, May 2004, Prague,

Scaramuzza, D., Martinelli, A. & Siegwart, R. (2006). A Toolbox for Easy calibrating Omnidirectional Cameras. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 5695–5701, China, October 2006, Beijing.

Scaramuzza, D., Martinelli, A. & Siegwart, R. (2006). A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion, *Proceedings of IEEE International Conference on Computer Vision Systems*, USA, January 2006, New York.

Levenberg, K. (1944). A Method for the Solution of Certain Problems in Least Squares, *Quarterly of Applied Mathematics*, Vol. 2, No. 2, pp. 164-168, July 1944.

Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM Journal on Applied Mathematics*. Vol. 11, No. 2, pp. 431-441, 1963.

Harris, C. & Stephens, M.J. (1988). A combined corner and edge detector, *Proceedings of The Fourth Alvey Vision Conference*, pp. 147-151, UK, 1988, Manchester.

Longuet-Higgins, H.C. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, Vol. 293, Sept 1981, pp. 133–135.

Vision Systems: Applications

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-01-1
Hard cover, 608 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

Computer Vision is the most important key in developing autonomous navigation systems for interaction with the environment. It also leads us to marvel at the functioning of our own vision system. In this book we have collected the latest applications of vision research from around the world. It contains both the conventional research areas like mobile robot navigation and map building, and more recent applications such as, micro vision, etc.The fist seven chapters contain the newer applications of vision like micro vision, grasping using vision, behavior based perception, inspection of railways and humanitarian demining. The later chapters deal with applications of vision in mobile robot navigation, camera calibration, object detection in vision search, map building, etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen