

A Generic Camera Model and Calibration Method for Conventional, Wide-Angle, and Fish-Eye Lenses

Juho Kannala, *Student Member, IEEE*, and
Sami S. Brandt, *Member, IEEE*

Abstract—Fish-eye lenses are convenient in such applications where a very wide angle of view is needed, but their use for measurement purposes has been limited by the lack of an accurate, generic, and easy-to-use calibration procedure. We hence propose a generic camera model, which is suitable for fish-eye lens cameras as well as for conventional and wide-angle lens cameras, and a calibration method for estimating the parameters of the model. The achieved level of calibration accuracy is comparable to the previously reported state-of-the-art.

Index Terms—Camera model, camera calibration, lens distortion, fish-eye lens, wide-angle lens.

1 INTRODUCTION

THE pinhole camera model accompanied with lens distortion models is a fair approximation for most conventional cameras with narrow-angle or even wide-angle lenses [1], [2], [3]. But, it is still not suitable for fish-eye lens cameras. Fish-eye lenses are designed to cover the whole hemispherical field in front of the camera and the angle of view is very large, about 180°. Moreover, it is impossible to project the hemispherical field of view on a finite image plane by a perspective projection, so fish-eye lenses are designed to obey some other projection model. This is the reason why the inherent distortion of a fish-eye lens should not be considered only as a deviation from the pinhole model [4].

There have been some efforts to model the radially symmetric distortion of fish-eye lenses with different models [5], [6], [7]. The idea in many of these approaches is to transform the original fish-eye image to follow the pinhole model. In [6] and [7], the parameters of the distortion model are estimated by forcing straight lines straight after the transformation, but the problem is that the methods do not give the full calibration. They can be used to “correct” the images to follow the pinhole model but their applicability is limited when one needs to know the direction of a back-projected ray corresponding to an image point. The calibration procedures in [8] and [9] instead aim at calibrating fish-eye lenses generally. However, these methods are slightly cumbersome, in practice, because a laser beam or a cylindrical calibration object is required.

Recently, the first autocalibration methods for fish-eye lens cameras have also emerged [10], [11], [12]. Mičušík and Pajdla [10] proposed a method for simultaneous linear estimation of epipolar geometry and an omnidirectional camera model. Claus and Fitzgibbon [11] presented a distortion model which likewise allows the simultaneous linear estimation of camera motion and lens geometry, and Thirthala and Pollefeys [12] used the multi-view-view geometry of radial 1D cameras to estimate a nonparametric camera model. In addition, the recent work by Barreto and

Daniilidis [13] introduced a radial fundamental matrix for correcting the distortion of wide-angle lenses. Nevertheless, the emphasis in these approaches is more in the autocalibration techniques than in the precise modeling of real lenses.

In this paper, we concentrate on accurate geometric modeling of real cameras.¹ We propose a novel calibration method for fish-eye lenses that requires that the camera observes a planar calibration pattern. The calibration method is based on a generic camera model that will be shown to be suitable for different kind of omnidirectional cameras as well as for conventional cameras. First, in Section 2, we present the camera model and, in Section 3, theoretically justify it by comparing different projection models. In Section 4, we describe a procedure for estimating the parameters of the camera model, and the experimental results are presented and discussed in Sections 5 and 6.

2 GENERIC CAMERA MODEL

Since the perspective projection model is not suitable for fish-eye lenses we use a more flexible radially symmetric projection model. This basic model is introduced in Section 2.1 and then extended with asymmetric distortion terms in Section 2.2. Computation of back-projections is described in Section 2.3.

2.1 Radially Symmetric Model

The perspective projection of a pinhole camera can be described by the following formula

$$r = f \tan \theta \quad (\text{i. perspective projection}), \quad (1)$$

where θ is the angle between the principal axis and the incoming ray, r is the distance between the image point and the principal point, and f is the focal length. Fish-eye lenses instead are usually designed to obey one of the following projections:

$$r = 2f \tan(\theta/2) \quad (\text{ii. stereographic projection}), \quad (2)$$

$$r = f\theta \quad (\text{iii. equidistance projection}), \quad (3)$$

$$r = 2f \sin(\theta/2) \quad (\text{iv. equisolid angle projection}), \quad (4)$$

$$r = f \sin(\theta) \quad (\text{v. orthogonal projection}). \quad (5)$$

Perhaps the most common model is the equidistance projection. The behavior of the different projections is illustrated in Fig. 1a and the difference between a pinhole camera and a fish-eye camera is shown in Fig. 1b.

The real lenses do not, however, exactly follow the designed projection model. From the viewpoint of automatic calibration, it would also be useful if we had only one model suitable for different types of lenses. Therefore, we consider projections in the general form

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + k_5\theta^9 + \dots, \quad (6)$$

where, without any loss of generality, even powers have been dropped. This is due to the fact that we may extend r onto the negative side as an odd function while the odd powers span the set of continuous odd functions. For computations, we need to fix the number of terms in (6). We found that first five terms, up to the ninth power of θ , give enough degrees of freedom for good approximation of different projection curves. Thus, the radially symmetric part of our camera model contains the five parameters, k_1, k_2, \dots, k_5 .

Let \mathcal{F} be the mapping from the incoming rays to the normalized image coordinates

1. An early conference version of this paper is [14].

• J. Kannala is with the Department of Electrical Engineering, University of Oulu, PO Box 4500, FIN-90014, Finland. E-mail: juho.kannala@ee.oulu.fi.
• S.S. Brandt is with the Laboratory of Computational Engineering, Helsinki University of Technology, PO Box 9203, FIN-02015 TKK, Finland. E-mail: sami.brandt@tkk.fi.

Manuscript received 3 Feb. 2005; revised 23 Sept. 2005; accepted 21 Nov. 2005; published online 13 June 2006.

Recommended for acceptance by K. Daniilidis.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0070-0205.

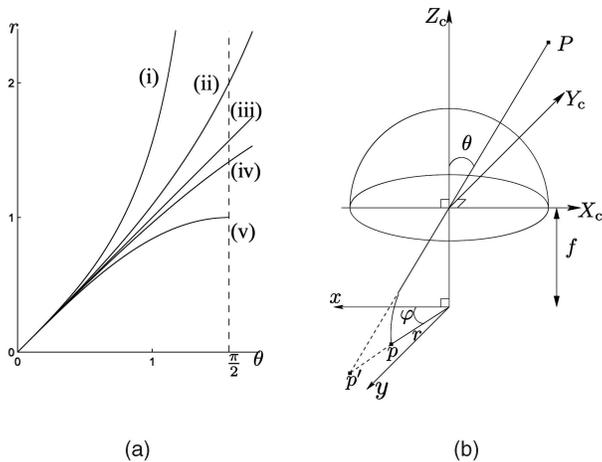


Fig. 1. (a) Projections (1), (2), (3), (4), and (5) with $f = 1$. (b) Fish-eye camera model. The image of the point P is p whereas it would be p' by a pinhole camera.

$$\begin{pmatrix} x \\ y \end{pmatrix} = r(\theta) \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} = \mathcal{F}(\Phi), \quad (7)$$

where $r(\theta)$ contains the first five terms of (6) and $\Phi = (\theta, \varphi)^\top$ is the direction of the incoming ray. For real lenses, the values of parameters k_i are such that $r(\theta)$ is monotonically increasing on the interval $[0, \theta_{\max}]$, where θ_{\max} is the maximum viewing angle. Hence, when computing the inverse of \mathcal{F} , we may solve θ by numerically finding the roots of a ninth order polynomial and then choosing the real root between 0 and θ_{\max} .

2.2 Full Model

Real lenses may deviate from precise radial symmetry and, therefore, we supplement our model with an asymmetric part. For instance, the lens elements may be inaccurately aligned causing that the projection is not exactly radially symmetric. With conventional lenses, this kind of distortion is called decentering distortion [1], [15]. However, there are also other possible sources of imperfections in the optical system and some of them may be difficult to model. For example, the image plane may be tilted with respect to the principal axis or the individual lens elements may not be precisely radially symmetric. Therefore, instead of trying to model all different physical phenomena in the optical system individually, we propose a flexible mathematical distortion model that is just fitted to agree with the observations.

To obtain a widely applicable, flexible model, we propose to use two distortion terms as follows: One distortion term acts in the radial direction

$$\Delta_r(\theta, \varphi) = (l_1\theta + l_2\theta^3 + l_3\theta^5)(i_1 \cos \varphi + i_2 \sin \varphi + i_3 \cos 2\varphi + i_4 \sin 2\varphi), \quad (8)$$

and the other in the tangential direction

$$\Delta_t(\theta, \varphi) = (m_1\theta + m_2\theta^3 + m_3\theta^5)(j_1 \cos \varphi + j_2 \sin \varphi + j_3 \cos 2\varphi + j_4 \sin 2\varphi), \quad (9)$$

where the distortion functions are separable in the variables θ and φ . Because the Fourier series of any 2π -periodic continuous function converges in the L^2 -norm and any continuous odd function can be represented by a series of odd polynomials we could, in principle, model any kind of continuous distortion by simply adding more terms to (8) and (9), as they both now have seven parameters.

By adding the distortion terms to (7), we obtain the distorted coordinates $\mathbf{x}_d = (x_d, y_d)^\top$ by

$$\mathbf{x}_d = r(\theta)\mathbf{u}_r(\varphi) + \Delta_r(\theta, \varphi)\mathbf{u}_r(\varphi) + \Delta_t(\theta, \varphi)\mathbf{u}_\varphi(\varphi), \quad (10)$$

where $\mathbf{u}_r(\varphi)$ and $\mathbf{u}_\varphi(\varphi)$ are the unit vectors in the radial and tangential directions. To achieve a complete camera model, we still need to transform the sensor plane coordinates into the image pixel coordinates. By assuming that the pixel coordinate system is orthogonal, we get the pixel coordinates $(u, v)^\top$ from

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} m_u & 0 \\ 0 & m_v \end{bmatrix} \begin{pmatrix} x_d \\ y_d \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \mathcal{A}(\mathbf{x}_d), \quad (11)$$

where $(u_0, v_0)^\top$ is the principal point and m_u and m_v give the number of pixels per unit distance in horizontal and vertical directions, respectively.

By combining (10) and (11), we have the forward camera model

$$\mathbf{m} = \mathcal{P}_c(\Phi), \quad (12)$$

where $\mathbf{m} = (u, v)^\top$. This full camera model contains 23 parameters and it is denoted by \mathbf{p}_{23} in the following. Since the asymmetric part of the model is very flexible, it may sometimes be reasonable to use a reduced camera model in order to avoid over-fitting. This is the case if, for instance, the control points do not cover the whole image area. Leaving out the asymmetric part gives the camera model \mathbf{p}_9 with nine parameters: five in the radially symmetric part (7) and four in the affine transformation (11). We did experiments also with the six-parametric model \mathbf{p}_6 which contains only two parameters in the radially symmetric part.

2.3 Backward Model

Above, we have described our forward camera model \mathcal{P}_c . In practice, one also needs to know the backward model

$$\Phi = \mathcal{P}_c^{-1}(\mathbf{m}), \quad (13)$$

which is the mapping from the image point $\mathbf{m} = (u, v)^\top$ to the direction of an incoming light ray, $\Phi = (\theta, \varphi)^\top$. We write \mathcal{P}_c as the composite function $\mathcal{P}_c = \mathcal{A} \circ \mathcal{D} \circ \mathcal{F}$, where \mathcal{F} is the transformation (7) from the ray direction Φ to the ideal Cartesian coordinates $\mathbf{x} = (x, y)^\top$ on the image plane, \mathcal{D} is the distortion mapping from \mathbf{x} to the distorted coordinates $\mathbf{x}_d = (x_d, y_d)^\top$ and \mathcal{A} is the affine transformation (11). We decompose the projection model in this form because, for the inverse transform $\mathcal{P}_c^{-1} = \mathcal{F}^{-1} \circ \mathcal{D}^{-1} \circ \mathcal{A}^{-1}$, it is straightforward to compute \mathcal{F}^{-1} and \mathcal{A}^{-1} . The more difficult part is to numerically compute \mathcal{D}^{-1} .

Given a point \mathbf{x}_d , finding $\mathbf{x} = \mathcal{D}^{-1}(\mathbf{x}_d)$ is equivalent to computing the shift \mathbf{s} into the expression $\mathbf{x} = \mathbf{x}_d - \mathbf{s}$, where

$$\mathbf{s} = \mathcal{S}(\Phi) = \Delta_r(\theta, \varphi)\mathbf{u}_r(\varphi) + \Delta_t(\theta, \varphi)\mathbf{u}_\varphi(\varphi). \quad (14)$$

Moreover, we may write $\mathcal{S}(\Phi) \equiv (\mathcal{S} \circ \mathcal{F}^{-1})(\mathbf{x})$ and approximate the shift by the first order Taylor expansion of $\mathcal{S} \circ \mathcal{F}^{-1}$ around \mathbf{x}_d that yields

$$\begin{aligned} \mathbf{s} &\simeq (\mathcal{S} \circ \mathcal{F}^{-1})(\mathbf{x}_d) + \frac{\partial(\mathcal{S} \circ \mathcal{F}^{-1})}{\partial \mathbf{x}}(\mathbf{x}_d)(\mathbf{x} - \mathbf{x}_d) \\ &= \mathcal{S}(\Phi_d) - \frac{\partial \mathcal{S}}{\partial \Phi} \left(\frac{\partial \mathcal{F}}{\partial \Phi}(\Phi_d) \right)^{-1} \mathbf{s}, \end{aligned}$$

where $\Phi_d = \mathcal{F}^{-1}(\mathbf{x}_d)$ may be numerically evaluated. Hence, we may compute the shift \mathbf{s} from

$$\mathbf{s} \simeq \left(I + \frac{\partial \mathcal{S}}{\partial \Phi}(\Phi_d) \left(\frac{\partial \mathcal{F}}{\partial \Phi}(\Phi_d) \right)^{-1} \right)^{-1} \mathcal{S}(\Phi_d), \quad (15)$$

where the Jacobians $\partial \mathcal{S} / \partial \Phi$ and $\partial \mathcal{F} / \partial \Phi$ may be computed from (14) and (7), respectively. So, finally,

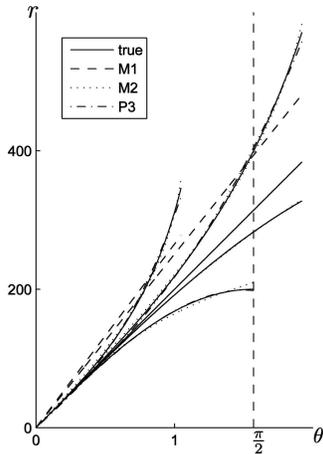


Fig. 2. Approximations.

$$\mathcal{D}^{-1}(\mathbf{x}_d) \simeq \mathbf{x}_d - \left(I + \left(\frac{\partial \mathcal{S}}{\partial \Phi} \circ \mathcal{F}^{-1} \right) (\mathbf{x}_d) \left(\left(\frac{\partial \mathcal{F}}{\partial \Phi} \circ \mathcal{F}^{-1} \right) (\mathbf{x}_d) \right)^{-1} \right)^{-1} (\mathcal{S} \circ \mathcal{F}^{-1})(\mathbf{x}_d). \quad (16)$$

It seems that the first order approximation for the asymmetric distortion function \mathcal{D} is tenable, in practice, because the backward model error is typically several degrees smaller than the calibration accuracy for the forward model, as will be seen in detail in Section 5.

3 JUSTIFICATION OF THE PROJECTION MODEL

The traditional approach for camera calibration is to take the perspective projection model as a starting point and then supplement it with distortion terms [1], [3], [16]. However, this is not a valid approach for fish-eye lenses because, when θ approaches $\pi/2$, the perspective model projects points infinitely far and it is not possible to remove this singularity with the conventional distortion models. Hence, we base our calibration method to the more generic model (6).

We compared the polynomial projection model (6) to the two two-parametric models

$$r = \frac{a}{b} \sin(b\theta) \quad (M1) \quad \text{and} \quad r = \frac{a - \sqrt{a^2 - 4b\theta^2}}{2b\theta} \quad (M2),$$

proposed by Mičušík [17] for fish-eye lenses. In Fig. 2, we have plotted the projection curves (1), (2), (3), (4), and (5) and their least-squares approximations with models $M1$, $M2$, and $P3$, where $P3$ is the polynomial model (6) with the first two terms. Here, we used the value $f = 200$ pixels which is a reasonable value for a real camera. The projections were approximated between 0 and θ_{\max} , where the values of θ_{\max} were 60° , 110° , 110° , 110° , and 90° , respectively. The interval $[0, \theta_{\max}]$ was discretized using the step of 0.1° and the models $M1$ and $M2$ were fitted by using the Levenberg-Marquardt method. It can be seen from Fig. 2 that the model $M1$ is not suitable at all for the perspective and stereographic projections and that the model $M2$ is not accurate for the orthogonal projection.

In Table 1, we have tabulated the maximum approximation errors for each model, i.e., the maximum vertical distances between the desired curve and the approximation in Fig. 2. Here, we also have the model $P9$ which is the polynomial model (6) with the first five terms. It can be seen that the model $P3$ has the best overall performance from all of the two-parametric models and that the sub-pixel approximation accuracy for all the projection curves

TABLE 1
The Approximation Errors

	$M1$	$M2$	$P3$	$P9$
(1)	69	13	12	0.1
(2)	90	13	13	0.0
(3)	0.0	0.0	0.0	0.0
(4)	0.0	1.8	0.33	0.0
(5)	0.0	9.7	1.80	0.0

requires the five-parametric model $P9$. These results show that the radially symmetric part of our camera model is well justified.

4 CALIBRATING THE GENERIC MODEL

Next, we describe a procedure for estimating the parameters of the camera model. The calibration method is based on viewing a planar object which contains control points in known positions. The advantage over the previous approaches is that also fish-eye lenses, possibly having a field of view larger than 180° , can be calibrated by simply viewing a planar pattern. In addition, a good accuracy can be achieved if circular control points are used, as described in Section 4.2.

4.1 Calibration Algorithm

The calibration procedure consists of four steps that are described below. We assume that M control points are observed in N views. For each view, there is a rotation matrix \mathbf{R}_j and a translation vector \mathbf{t}_j describing the position of the camera with respect to the calibration plane such that

$$\mathbf{X}_c = \mathbf{R}_j \mathbf{X} + \mathbf{t}_j, \quad j = 1, \dots, N. \quad (17)$$

We choose the calibration plane to lie in the XY -plane and denote the coordinates of the control point i with $\mathbf{X}^i = (X^i, Y^i, 0)^\top$. The corresponding homogeneous coordinates in the calibration plane are denoted by $\mathbf{x}_p^i = (X^i, Y^i, 1)^\top$ and the observed coordinates in the view j by $\mathbf{m}_j^i = (u_j^i, v_j^i)^\top$. The first three steps of the calibration procedure involve only six internal camera parameters and for these we use the short-hand notation $\mathbf{p}_6 \triangleq (k_1, k_2, m_u, m_v, u_0, v_0)$. The additional parameters of the full model are inserted only in the final step.

Step 1: Initialization of internal parameters. The initial guesses for k_1 and k_2 are obtained by fitting the model $r = k_1\theta + k_2\theta^3$ to the desired projection, (1)-(5), with the manufacturer's values for the nominal focal length f and the angle of view θ_{\max} . Then, we also obtain the radius of the image on the sensor plane by $r_{\max} = k_1\theta_{\max} + k_2\theta_{\max}^3$.

With a circular image fish-eye lens, the actual image fills only a circular area inside the image frames. In pixel coordinates, this circle is an ellipse

$$\left(\frac{u - u_0}{a} \right)^2 + \left(\frac{v - v_0}{b} \right)^2 = 1,$$

whose parameters can be estimated. Consequently, we obtain initial guesses for the remaining unknowns m_u , m_v , u_0 , and v_0 in \mathbf{p} , where $m_u = a/r_{\max}$ and $m_v = b/r_{\max}$. With a full-frame lens, the best thing is probably to place the principal point to the image center and use the reported values of the pixel dimensions to obtain initial values for m_u and m_v .

Step 2: Back-projection and computation of homographies. With the internal parameters \mathbf{p}_6 , we may back-project the observed

points \mathbf{m}_j^i onto the unit sphere centered at the camera origin (see Fig. 1b). The points on the sphere are denoted by $\tilde{\mathbf{x}}_j^i$. Since the mapping between the points on the calibration plane and on the unit sphere is a central projection, there is a planar homography \mathbf{H}_j so that $s\tilde{\mathbf{x}}_j^i = \mathbf{H}_j\mathbf{x}_p^i$.

For each view j the homography \mathbf{H}_j is computed as follows:

1. Back-project the control points by first computing the normalized image coordinates

$$\begin{pmatrix} x_j^i \\ y_j^i \end{pmatrix} = \begin{bmatrix} 1/m_u & 0 \\ 0 & 1/m_v \end{bmatrix} \begin{pmatrix} u_j^i - u_0 \\ v_j^i - v_0 \end{pmatrix},$$

transforming them to the polar coordinates $(r_j^i, \varphi_j^i) = (x_j^i, y_j^i)$ and, finally, solving θ_j^i from the cubic equation $k_2(\theta_j^i)^3 + k_1\theta_j^i - r_j^i = 0$.

2. Set $\tilde{\mathbf{x}}_j^i = (\sin \varphi_j^i \sin \theta_j^i, \cos \varphi_j^i \sin \theta_j^i, \cos \theta_j^i)$.
3. Compute the initial estimate for \mathbf{H}_j from the correspondences $\tilde{\mathbf{x}}_j^i \leftrightarrow \mathbf{x}_p^i$ by the linear algorithm with data normalization [18]. Define $\hat{\mathbf{x}}_j^i$ as the exact image of \mathbf{x}_p^i under \mathbf{H}_j such that $\hat{\mathbf{x}}_j^i = \mathbf{H}_j\mathbf{x}_p^i / \|\mathbf{H}_j\mathbf{x}_p^i\|$.
4. Refine the homography \mathbf{H}_j by minimizing $\sum_i \sin^2 \alpha_j^i$, where α_j^i is the angle between the unit vectors $\tilde{\mathbf{x}}_j^i$ and $\hat{\mathbf{x}}_j^i$.

Step 3: Initialization of external parameters. The initial values for the external camera parameters are extracted from the homographies \mathbf{H}_j . It holds that

$$s\tilde{\mathbf{x}}_j^i = [\mathbf{R}_j \quad \mathbf{t}_j] \begin{pmatrix} X^i \\ Y^i \\ 0 \\ 1 \end{pmatrix} = [\mathbf{r}_j^1 \quad \mathbf{r}_j^2 \quad \mathbf{t}_j] \begin{pmatrix} X^i \\ Y^i \\ 1 \end{pmatrix},$$

which implies $\mathbf{H}_j = [\mathbf{r}_j^1 \quad \mathbf{r}_j^2 \quad \mathbf{t}_j]$, up to scale. Furthermore,

$$\mathbf{r}_j^1 = \lambda_j \mathbf{h}_j^1, \quad \mathbf{r}_j^2 = \lambda_j \mathbf{h}_j^2, \quad \mathbf{r}_j^3 = \mathbf{r}_j^1 \times \mathbf{r}_j^2, \quad \mathbf{t}_j = \lambda_j \mathbf{h}_j^3,$$

where $\lambda_j = \text{sign}(H_j^{3,3}) / \|\mathbf{h}_j^1\|$. Because of estimation errors, the obtained rotation matrices are not orthogonal. Thus, we use the singular value decomposition to compute the closest orthogonal matrices in the sense of Frobenius norm [19] and use them as initial guess for each \mathbf{R}_j .

Step 4: Minimization of projection error. If the full model \mathbf{p}_{23} or the model \mathbf{p}_9 is used the additional camera parameters are initialized to zero at this stage. As we have the estimates for the internal and external camera parameters, we use (17), (7) or (10), and (11) to compute the imaging function \mathcal{P}_j for each camera, where a control point is projected to $\hat{\mathbf{m}}_j^i = \mathcal{P}_j(\mathbf{X}^i)$. The camera parameters are refined by minimizing the sum of squared distances between the measured and modeled control point projections

$$\sum_{j=1}^N \sum_{i=1}^M d(\mathbf{m}_j^i, \hat{\mathbf{m}}_j^i)^2 \quad (18)$$

using the Levenberg-Marquardt algorithm.

4.2 Modification for Circular Control Points

In order to achieve an accurate calibration, we used a calibration plane with white circles on black background since the centroids of the projected circles can be detected with a subpixel level of accuracy [20]. In this setting, however, the problem is that the centroid of the *projected* circle is not the image of the center of the original circle. Therefore, since \mathbf{m}_j^i in (18) is the measured centroid, we should not project the centers as points $\hat{\mathbf{m}}_j^i$.

To avoid the problem above, we propose solving the centroids of the projected circles numerically. We parameterize the interior of the circle at (X_0, Y_0) with radius R by

TABLE 2
The RMS Residual Error in Pixels

	δ_8	\mathbf{p}_6	\mathbf{p}_9	\mathbf{p}_{23}
Cosmicar	0.061	0.107	0.055	0.052
Sony	0.124	0.234	0.092	0.057

$\mathbf{X}(\varrho, \alpha) = (X_0 + \varrho \sin \alpha, Y_0 + \varrho \cos \alpha, 0)^\top$. Given the camera parameters, we get the centroid $\hat{\mathbf{m}}$ for the circle by numerically evaluating

$$\hat{\mathbf{m}} = \frac{\int_0^R \int_0^{2\pi} \hat{\mathbf{m}}(\varrho, \alpha) |\det \mathbf{J}(\varrho, \alpha)| d\alpha d\varrho}{\int_0^R \int_0^{2\pi} |\det \mathbf{J}(\varrho, \alpha)| d\alpha d\varrho}, \quad (19)$$

where $\hat{\mathbf{m}}(\varrho, \alpha) = \mathcal{P}(\mathbf{X}(\varrho, \alpha))$ and $\mathbf{J}(\varrho, \alpha)$ is the Jacobian of the composite function $\mathcal{P} \circ \mathbf{X}$. The analytical solving of the Jacobian is a rather tedious task, but it can be computed by mathematical software such as Maple.

5 CALIBRATION EXPERIMENTS

5.1 Conventional and Wide-Angle Lens Camera

The proposed camera model was compared to the camera model used by Heikkilä [3]. This model is the skew-zero pinhole model accompanied with four distortion parameters and it is denoted by δ_8 in the following.

In the first experiment, we used the same data, provided by Heikkilä, as in [3]. It was originally obtained by capturing a single image of a calibration object consisting of two orthogonal planes, each with 256 circular control points. The camera was a monochrome CCD camera with a 8.5 mm Cosmicar lens. The second experiment was performed with the Sony DFW-VL500 camera and a wide-angle conversion lens, with total focal length of 3.8 mm. In this experiment, we used six images of the calibration object. There were 1,328 observed control points in total and they were localized by computing their gray-scale centroids [20].

The obtained RMS residual errors, i.e., the root-mean-squared distances between the measured and modeled control point positions, are shown in Table 2. Especially interesting is the comparison between models δ_8 and \mathbf{p}_9 because they both have eight degrees of freedom. Model \mathbf{p}_9 gave slightly smaller residuals although it does not contain any tangential distortion terms. The full model \mathbf{p}_{23} gave the smallest residuals.

However, in the first experiment the full model may have been partly fitted to the systematic errors of the calibration data. This is due to the fact that there were measurements only from one image where the illumination was not uniform and all corners were not covered by control points. To illustrate the fact, the estimated asymmetric distortion and remaining residuals for the model \mathbf{p}_{23} are shown in Fig. 3. The relatively large residuals in the lower right corner of the calibration image (Fig. 3b) seem to be due to inaccurate localization, caused by nonuniform lighting.

In the second experiment, the calibration data was better, so the full model is likely to be more useful. This was verified by taking an additional image of the calibration object and solving the corresponding external camera parameters with given internal parameters. The RMS projection error for the additional image was 0.049 pixels for \mathbf{p}_{23} and 0.071 for \mathbf{p}_9 . This indicates that the full model described the true geometry of the camera better than the simpler model \mathbf{p}_9 .

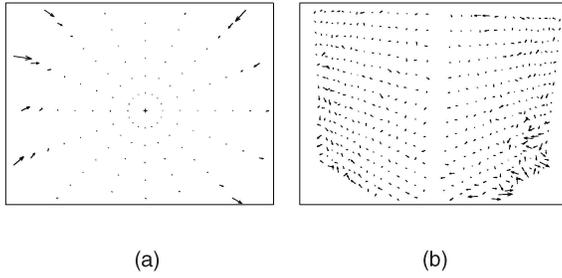


Fig. 3. Heikkilä's calibration data. (a) The estimated asymmetric distortion ($\Delta_r u_r + \Delta_t u_t$) using the extended model p_{23} . (b) The remaining residual for each control point. The vectors are scaled up by a factor of 150.

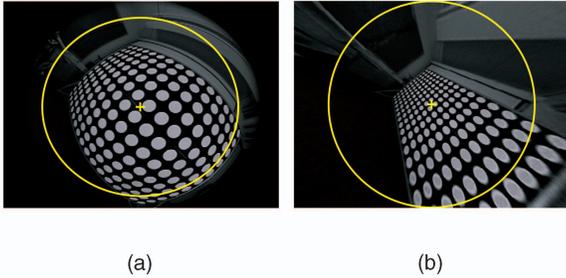


Fig. 4. Fish-eye lens calibration using only one view. (a) Original image where the ellipse depicts the field of view of 150° . (b) The image corrected to follow pinhole model. Straight lines are straight as they should be.

TABLE 3
The RMS Residual Error in Pixels

	p_6	p_9	p_{23}
Watec	0.146	0.094	0.089
ORIFL	0.491	0.167	0.137

Finally, we estimated the backward model error for p_{23} , caused by the first order approximation of the asymmetric distortion function (see Section 2.3). This was done by back-projecting each pixel and then reprojecting the rays. The maximum displacement in the reprojection was $2.1 \cdot 10^{-5}$ pixels for the first camera and $4.6 \cdot 10^{-4}$ pixels for the second. Both values are very small so it is justified to ignore the backward model error in practice.

5.2 Fish-Eye Lens Cameras

The first experimented fish-eye lens was an equidistance lens with the nominal focal length of 1.178 mm and it was attached to a Watec 221S CCD color camera. The calibration object was a 2×3 m² plane containing white circles with the radius of 60 mm on the black background. The calibration images were digitized from an analog video signal to 8-bit monochrome images, whose size was 640 by 480 pixels.

The calibration of a fish-eye lens can be performed even from a single image of the planar object as Fig. 4 illustrates. In that example, we used the model p_6 and 60 control points. However, for the most accurate results, the whole field of view should be covered with a large number of measurements. Therefore, we experimented our method with 12 views and 680 points in total; the results are in Table 3. The extended model p_{23} had the smallest residual error but the radially symmetric model p_9 gave almost as good results. Nevertheless, there should be no risk of overfitting because the number of measurements is large. The estimated asymmetric distortion and the residuals are displayed in Fig. 5.

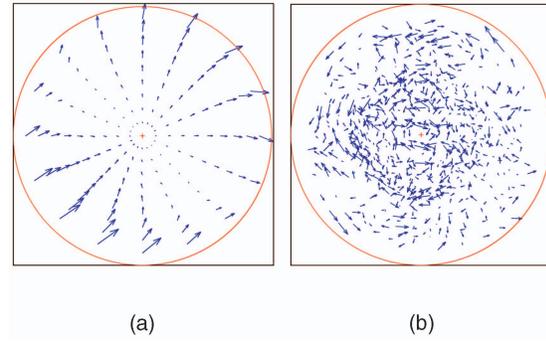


Fig. 5. (a) The estimated asymmetric distortion ($\Delta_r u_r + \Delta_t u_t$) using the extended model p_{23} . (b) The remaining residual for each control point that shows no obvious systematic error. Both plots are in normalized image coordinates and the vectors are scaled up by a factor of 150 to aid inspection.

The second fish-eye lens was ORIFL190-3 lens manufactured by Omnitech Robotics. This lens has a 190 degree field of view and it clearly deviates from the exact equidistance projection model. The lens was attached to a Point Grey Dragonfly digital color camera having $1,024 \times 768$ pixels; the calibration object was the same as in Section 5.1. The obtained RMS residual errors for a set-up of 12 views and 1,780 control points are shown in Table 3. Again, the full model had the best performance and this was verified with an additional calibration image. The RMS projection error for the additional image, after fitting the external camera parameters, was 0.13 pixels for p_{23} and 0.16 pixels for p_9 . The backward model error for p_{23} was evaluated at each pixel within the circular images. The maximum displacement was $9.7 \cdot 10^{-6}$ pixels for the first camera and $3.4 \cdot 10^{-3}$ pixels for the second. Again, it is justified to ignore such small errors in practice.

5.3 Synthetic Data

In order to evaluate the robustness of the proposed calibration method we did experiments also with synthetic data. The ground truth values for the camera parameters were obtained from the real fish-eye lens experiment that was illustrated in Fig. 5. So, we used the full camera model and we had 680 circular control points in 12 synthetic calibration images, where the gray level values of control points and background were 180 and 5, respectively. In order to make the synthetic images to better correspond real images, they were blurred by a Gaussian pdf ($\sigma = 1$ pixel) and quantized to the 256 gray levels.

First, we estimated the significance of the centroid correction proposed in Section 4.2. In the above setting the RMS distance between the centroids of the projected circles and the projected centers of the original circles was 0.45 pixels. It is a significantly larger value than the RMS residual errors reported in the real experiment (Table 3). This indicates that, without the centroid correction, the estimated camera parameters would have been biased and it is likely that the residual error would have been larger.

Second, we estimated the effect of noise to the calibration by adding Gaussian noise to the synthetic images and performing 10 calibration trials at each noise level. The standard deviation of the noise varied between 0 and 15 pixels. The control points were localized from the noisy images by first thresholding them using a fixed threshold. Then, the centroid of each control point was measured by computing the gray-level-weighted center-of-mass.

The simulation results are shown in Fig. 6, where we have plotted the average RMS measurement, RMS residual and RMS estimation errors. There is small error also at the zero noise level because of the discrete pixel representation and gray level quantization. The fact that the RMS errors approximately satisfy

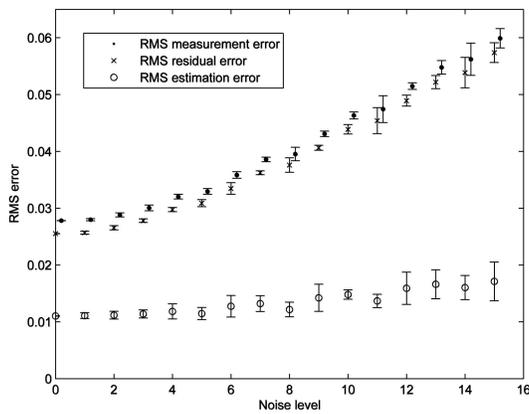


Fig. 6. The average RMS measurement, residual, and estimation errors for 10 calibration trials at different levels of noise. The errorbars represent the minimum and maximum values among the trials.

the Pythagorean equality indicates that the calibration algorithm has converged to the true global minimum [18]. Moreover, the low values of the RMS estimation error indicate that the estimated camera model is close to the true one even at large noise levels.

6 CONCLUSION

We have proposed a novel camera calibration method for fish-eye lens cameras that is based on viewing a planar calibration pattern. The experiments verify that the method is easy-to-use and provides a relatively high level of accuracy with circular control points. The proposed camera model is generic, easily expandable, and suitable also for conventional cameras with narrow or wide-angle lenses. The achieved level of accuracy for fish-eye lenses is better than it has been reported with other approaches and, for narrow-angle lenses, it is comparable to the results in [3]. This is promising considering especially the aim of using fish-eye lenses in measurement purposes. The calibration method is implemented as a calibration toolbox on Matlab and is available on the authors' Web page.

ACKNOWLEDGMENTS

The authors would like to thank Janne Heikkilä for discussions and for providing his data for the experiments.

REFERENCES

- [1] D.C. Brown, "Close-Range Camera Calibration," *Photogrammetric Eng.*, vol. 37, no. 8, pp. 855-866, 1971.
- [2] R. Swaminathan and S.K. Nayar, "Nonmetric Calibration of Wide-Angle Lenses and Polycameras," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1172-1178, Oct. 2000.
- [3] J. Heikkilä, "Geometric Camera Calibration Using Circular Control Points," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066-1077, Oct. 2000.
- [4] K. Miyamoto, "Fish Eye Lens," *J. Optical Soc. Am.*, vol. 54, no. 8, pp. 1060-1061, 1964.
- [5] A. Basu and S. Licardie, "Alternative Models for Fish-Eye Lenses," *Pattern Recognition Letters*, vol. 16, pp. 433-441, 1995.
- [6] F. Devernay and O. Faugeras, "Straight Lines Have to Be Straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14-24, 2001.
- [7] C. Bräuer-Burchardt and K. Voss, "A New Algorithm to Correct Fish-Eye and Strong Wide-Angle-Lens-Distortion from Single Images," *Proc. Int'l Conf. Image Processing*, pp. 225-228, 2001.
- [8] S. Shah and J. Aggarwal, "Intrinsic Parameter Calibration Procedure for a (High-Distortion) Fish-Eye Lens Camera with Distortion Model and Accuracy Estimation," *Pattern Recognition*, vol. 29, no. 11, pp. 1775-1788, 1996.
- [9] H. Bakstein and T. Pajdla, "Panoramic Mosaicing with a 180° Field of View Lens," *Proc. IEEE Workshop Omnidirectional Vision*, pp. 60-67, 2002.

- [10] B. Mičušík and T. Pajdla, "Estimation of Omnidirectional Camera Model from Epipolar Geometry," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 485-490, 2003.
- [11] D. Claus and A.W. Fitzgibbon, "A Rational Function Lens Distortion Model for General Cameras," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 213-219, 2003.
- [12] S. Thirithala and M. Pollefeys, "Multi-View Geometry of 1D Radial Cameras and Its Application to Omnidirectional Camera Calibration," *Proc. Int'l Conf. Computer Vision*, pp. 1539-1546, 2005.
- [13] J.P. Barreto and K. Daniilidis, "Fundamental Matrix for Cameras with Radial Distortion," *Proc. Int'l Conf. Computer Vision*, pp. 625-632, 2005.
- [14] J. Kannala and S. Brandt, "A Generic Camera Calibration Method for Fish-Eye Lenses," *Proc. 17th Int'l Conf. Pattern Recognition*, pp. 10-13, 2004.
- [15] A. Conrady, "Decentering Lens Systems," *Monthly Notices of the Royal Astronomical Soc.*, vol. 79, pp. 384-390, 1919.
- [16] *Manual of Photogrammetry*, C.C. Slama, ed., Am. Soc. Photogrammetry, 1980.
- [17] B. Mičušík, "Two-View Geometry of Omnidirectional Cameras," PhD dissertation, Czech Technical Univ., 2004.
- [18] R. Hartley and A. Zisserman, *Multiple View Geometry*, second ed. Cambridge, 2003.
- [19] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000.
- [20] J. Heikkilä and O. Silvén, "Calibration Procedure for Short Focal Length Off-the-Shelf CCD Cameras," *Proc. Int'l Conf. Pattern Recognition*, pp. 166-170, 1996.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.