

A New Non-central Model for Fisheye Calibration

Radka Tezaur Avinash Kumar Oscar Nestares
Intel Corp., Santa Clara, CA

radka.tezaur@intel.com, avinash1.kumar@intel.com, oscar.nestares@intel.com

Abstract

A new non-central model suitable for calibrating fisheye cameras is proposed. It is a direct extension of the popular central model developed by Scaramuzza et al., used by Matlab Computer Vision Toolbox fisheye calibration tool. It allows adapting existing applications that are using this central model to a non-central projection that is more accurate, especially when objects captured in the images are close to the camera, and it makes it possible to switch easily between the more accurate non-central characterization of the fisheye camera and the more convenient central approximation, as needed. It is shown that the algorithms proposed by Scaramuzza et al. for their central model can be modified to accommodate the angle dependent axial viewpoint shift. This means, besides other, that a similar process can be used for calibration involving the viewpoint shift characterization and a user-friendly calibration tool can be produced with this new non-central model that does not require the user to provide detailed lens design specifications or an educated guess for the initial parameter values. Several other improvements to the Scaramuzza's central model are also introduced, helping to improve the performance of both the central model, and its non-central extension.

1. Introduction

Fisheye lenses are attractive for many applications because of their huge field of view. In past their use was limited by the large image distortion that they cause. But as image data processing has become faster and cheaper, this no longer is a major obstacle and their use has become increasingly popular. They are now commonly used in security surveillance systems, automotive applications, robotics, 360° immersive still image and video capture devices, etc.

For performing 3D computer vision and computational photography tasks, camera systems need to be carefully calibrated. However, traditional methods used for calibrating regular cameras are generally not suitable for fisheye lenses with their large geometric distortion and the field of view that can exceed 180°. A number of special mod-



Figure 1. Our 360° camera array comprising 16 fisheye cameras

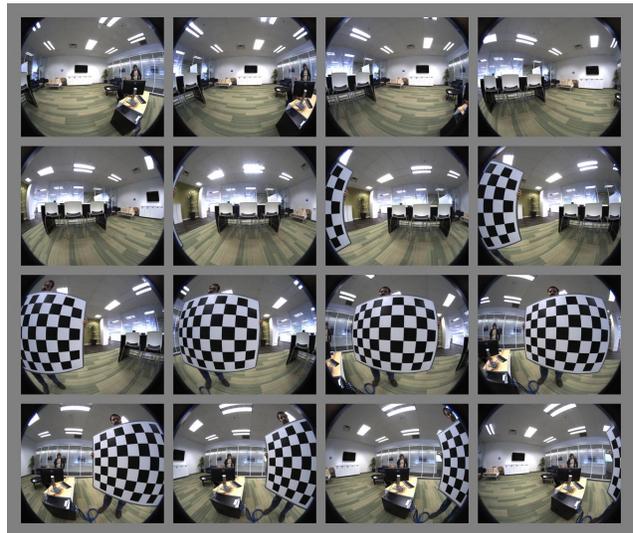


Figure 2. An example of the set of 16 images captured simultaneously by our 360° fisheye camera array, depicting the calibration chart at a single position.

els and algorithms thus have been developed for calibrating fisheye and other omnidirectional cameras. See, for example, [4, 18, 19, 24, 25, 30, 32, 38] and the references within.

One of the most popular tools for calibrating fisheye cameras is the one included in Matlab Computer Vision Toolbox, introduced in version 2017b, [1]. It is based on

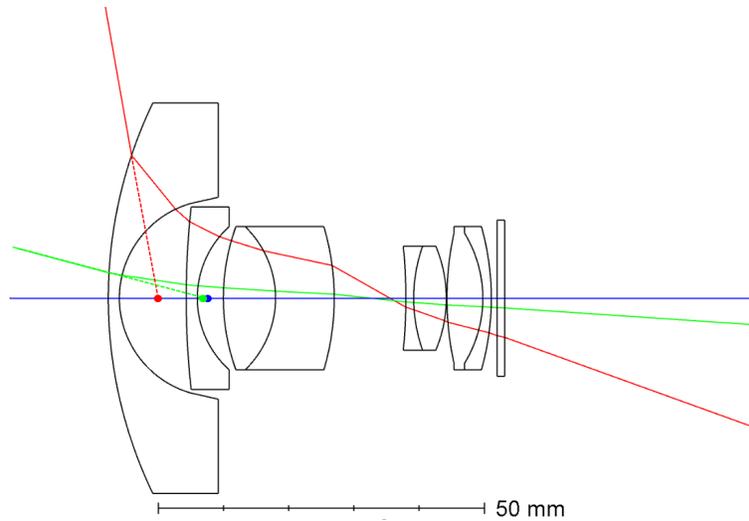


Figure 3. Nikon 16mm f/2.8D fisheye lens – apparent viewpoint shift between 0° (blue), 15° (green), and 80° (red) incident angle.

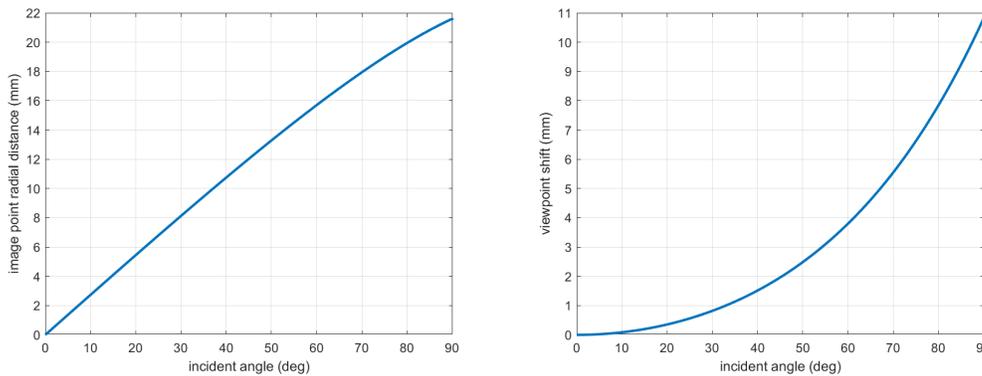


Figure 4. Characteristics of Nikon 16mm f/2.8D fisheye lens obtained from the lens design by ray tracing: image point radial distance (left) and apparent viewpoint shift amount (right) as the function of ray incident angle.

the OCamCalib package [31] developed by Scaramuzza *et al.*, [32, 33], and some of its later modifications, such as [37]. Apart from its wide availability, one of the advantages of this tool is that it is very easy to use. To calibrate a fisheye camera, the user only needs to capture the series of images of a common checkerboard chart and enter information about its size, which is readily available. It is not necessary to provide any detailed optical design specifications that might be hard to obtain, or any educated guess for the initial values of the calibration parameters, which might require some special knowledge and experience.

The vast majority of existing fisheye models, including those used by Matlab [1], OpenCV [2], and MPEG [39], assume a central projection. I.e., they assume that there exists a single viewpoint through which all the rays forming the image are passing. However, in practice, cameras equipped with fisheye lenses tend to be axial cameras. The entrance

pupil of the lens with extremely large field of view moves as the incident ray angle changes, which causes the apparent viewpoint to shift along the optical axis. This is shown shown, *e.g.*, in [35]. See also Figure 3 and 4. This viewpoint shift tends to be larger than many people realize. For example, in case of both Nikon fisheye lenses analyzed in [35], the apparent viewpoint shift between 0° and 90° incident angle exceeds 1 cm. The Fujinon fisheye lenses used in our 360° camera array, shown in Figure 1, also exhibit a viewpoint shift of a similar magnitude.

It is convenient to neglect this angle dependent viewpoint shift and use a central projection to model the camera. However, for some 3D computer vision and computational imaging applications this approximation may not be accurate enough. It can lead to large errors that potentially can limit system performance, especially when dealing with objects that are close to the cameras. For example, in case of

an object placed 0.5 m from the camera, neglecting 1 cm viewpoint shift for incident angles close to 90° results in more than 1° error in the assumed incident ray direction. When using cameras with high resolution sensors such as those used in Facebook 360° camera array described in [29], which provide angular resolution of about 35 pixels per degree, this translates to reprojection error in the order of tens of pixels. We believe that this is one of the reasons why Pozo *et al.* in their paper [29] report that their depth estimation starts to break down for objects closer than 1.6 m. For some more detailed results analyzing the impact of neglecting the axial viewpoint shift on the accuracy of some common stereo vision tasks see, e.g., [20, 21]. Note that the calibration with a chart belongs to tasks where neglecting the viewpoint shift of fisheye lenses may have significant impact on the accuracy. Due to the very large camera field of view, the calibration chart needs to be quite close to the camera. Otherwise it would have to be of size that is not practical. So, the calibration with a chart relies on spatial points that are most impacted by the parallax due to the angle-dependent viewpoint shift.

To mitigate this issue, several different non-central fish-eye and omnidirectional camera models have been proposed in literature. See, e.g., the work by Mičušík and Pajdla [26], Gonçalves and Araujo [14], Gennery [13], Swaminathan *et al.* [34], Kumar and Ahuja [22], Brousseau and Roy [5], Camposeco *et al.* [8], or Fasogbon and Aksu [11]. In this paper, we describe an alternative model that is a direct extension of Scaramuzza’s central model used by Matlab Computer Vision Toolbox. It makes it possible to adapt any existing applications that rely on this popular fisheye model to include also the viewpoint shift characterization and hence improve their performance when objects close to the camera are present in the scene. This applies, besides other, to the calibration tool itself. Our modification of Scaramuzza’s algorithms allows producing an easy-to-use fully automatic tool that can provide more accurate characterization of fisheye cameras, including their inherent incident angle dependent axial viewpoint shift. This is quite important because the viewpoint shift data, or the complete lens design from which this information can be extracted, is not something that is commonly provided by lens manufacturers and available to users.

Besides the extension of the model incorporating the axial viewpoint shift, we discuss in this paper also some other, smaller modifications of Scaramuzza’s central model that can help to improve the accuracy of the obtained calibration. Our experiments with both synthetic data, for which the ground truth is available, and with real life fisheye images demonstrate that the improvement in the accuracy achieved by using both the proposed modifications of the central model, and the extension to the non-central projection, can be quite significant.

2. Modified Scaramuzza’s central model

The omnidirectional camera model developed by Scaramuzza *et al.*, described in [32, 33], has two parts. The first one represents the central projection from 3D space to an ideal image plane involving radial distortion, while the second part is an affine transform representing the mapping from the ideal image plane to the captured image pixel coordinates. Together they use nine intrinsic parameters to characterize a fisheye camera. Four of them are the coefficients of polynomial

$$f(\rho) = f_0 + f_2\rho^2 + f_3\rho^3 + f_4\rho^4, \quad (1)$$

which represents the radial distortion. For any given point (x, y) in the ideal image plane it specifies a ray in 3D

$$d(x, y, f(\rho)), \quad d > 0, \quad (2)$$

that gets projected to (x, y) . Here and elsewhere ρ denotes the radial distance $\rho = \sqrt{x^2 + y^2}$ from the principal point – the point where optical axis z intersects the image plane, which is the center of the radial distortion. The five-parameter affine transform that maps the point (x, y) in the ideal image plane to a fisheye image point (u, v) has the form

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}. \quad (3)$$

Note that the polynomial (1) does not include a linear term. The argument for omitting this term can be extended to all odd terms. This is similar to, e.g., the traditional Brown distortion model [6, 7] using only even terms. We thus prefer to replace (1) by

$$f(\rho) = \sum_{k=0}^K f_{2k}\rho^{2k}. \quad (4)$$

Our experiments show that the polynomial (4) with four free parameters tends to yield more accurate calibration than (1) and that using polynomials of higher order can further improve the accuracy of calibration, see Section 5.

Also, it should be noted that one of the parameters in the affine transform (3) is redundant and can be eliminated. The coordinates in the ideal image plane can always be rotated and scaled in such a way that a_3 (or possibly a_2) becomes 0. In our experiments removing this parameter both eliminated small random rotations in estimated camera extrinsics, which can be misleading, and improved the convergence of the final non-linear optimization step. Similarly as Urban *et al.* in [37] we observe that the extra step involving a brute force search for more accurate initial estimate of the principal point (c_1, c_2) , included in the original OCam-Calib package, performed between the initial estimation of

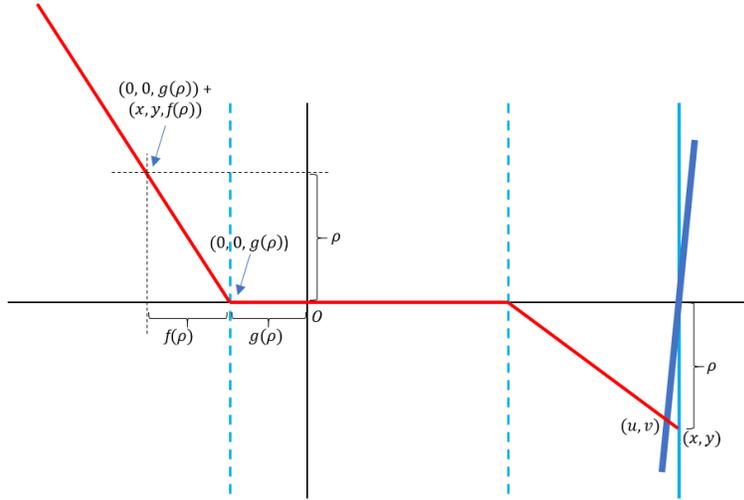


Figure 5. Proposed non-central extension of Scaramuzza's fisheye model involving the characterization of axial viewpoint shift expressed as the function of image point radial distance ρ .

distortion and extrinsics and the final non-linear optimization, see [33], is unnecessary with our modifications.

Similarly as, e.g., Frank *et al.* [12], we also observe that in the presence of radial distortion the mapping from an ideal image plane to a tilted sensor plane cannot be reduced to an affine transform, as it is commonly done with the pinhole camera model (see, e.g., [15, 28]). A projective transform in this case provides more physically accurate model for a sensor that is not perfectly aligned to the lens. A general homography representing the projective transform between two planes in 3D has eight free parameters. However, here two of them are redundant. The homography between the ideal image plane and camera sensor plane can be parameterized by six parameters as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & c_1 \\ 0 & 1 & c_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ p_1 & p_2 & 1 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

The factorization above not only provides non-redundant parameterization but also is convenient for implementation, as it allows implementing both the forward and inverse transform in a few simple steps. The homography matrix representing the transform from the image to the ideal image plane can be expressed as

$$\mathbf{H}^{-1} = \begin{bmatrix} \frac{1}{a_1} & \frac{-a_2}{a_1} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -p_1 & -p_2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -c_1 \\ 0 & 1 & -c_2 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

Note that when the redundant parameter a_3 is normalized to 0, the affine transform parameterization used by Scaramuzza's model is the special case of our more general projective transform parameterization (5), with $p_1 = p_2 = 0$.

Our proposed projective transform is thus a direct extension of the affine transform used in Scaramuzza's model used by Matlab Computer Vision Toolbox, adding two extra free parameters.

3. Proposed extension to non-central model

To extend either the original Scaramuzza's model, or our preferred central model with the modifications described in Section 2 to a non-central axial projection, we introduce the second polynomial $g(\rho)$ that characterizes the amount of viewpoint shift along the optical axis. With this new model, instead of (2), all points that get projected to the point (x, y) in the ideal image plane form the half-line

$$(0, 0, g(\rho)) + d(x, y, f(\rho)), \quad d > 0. \quad (7)$$

I.e., the ray that gets projected to point (x, y) starts not at the origin, but at the point $(0, 0, g(\rho))$ on the optical axis, see Figure 5.

By a similar argument as in the case of radial distortion polynomial f , the newly introduced function g that characterizes the amount viewpoint shift should be a smooth, even function. Also, without the loss of generality, we can choose the origin of the camera coordinate system to be at the viewpoint corresponding to zero incident angle, which makes $g(0) = 0$. Consequently, we require that

$$g(\rho) = \sum_{m=1}^M g_{2m} \rho^{2m}. \quad (8)$$

The key difference compared to some other non-central models, such as those presented in [13] and [22], is that we

express the amount of the shift of the apparent viewpoint along the optical axis not as the function of incident angle, but as the function of the radial distance of the image point from the principal point. When using the other models to find the projected image point, it is necessary first to find the incident angle. However this operation is non-trivial, as the incident angle depends on the amount of viewpoint shift, but at the same time the amount viewpoint shift is expressed as the function of incident angle. Finding the viewpoint and the incident angle thus requires solving a non-linear equation. With our model, this is not necessary.

Given a 3D point (X, Y, Z) that should be projected to the image, with our new model we have

$$(X, Y, Z) = (0, 0, g(\rho)) + d(x, y, f(\rho)) \quad (9)$$

for some x, y , and $d > 0$. Comparing the individual components and denoting $R = \sqrt{X^2 + Y^2}$ yields

$$d = \frac{X}{x} = \frac{Y}{y} = \frac{Z - g(\rho)}{f(\rho)} = \frac{R}{\rho}. \quad (10)$$

We can thus obtain ρ by solving the equation

$$f(\rho) - \frac{Z}{R}\rho + \frac{1}{R}g(\rho) = 0, \quad (11)$$

then find the corresponding point (x, y) in the ideal image plane,

$$x = \frac{\rho}{R}X, \quad y = \frac{\rho}{R}Y, \quad (12)$$

and finally find the projected image point (u, v) by applying to (x, y) the affine transform (3) or our homography (5).

The mapping in the opposite direction, identifying a 3D point or the ray in 3D corresponding to the image point (u, v) is similarly as in the case of Scaramuzza's central model trivial and requires only evaluating a few formulas. It starts with finding the corresponding point (x, y) in the ideal image plane by applying the inverse affine transform or inverse homography (6) to the image point (u, v) . The ray in 3D is then given by (7) together with (4).

4. Calibration algorithm

4.1. Initial guess for the calibration parameters

The algorithm for estimating initial values of calibration parameters described by Scaramuzza *et al.* in [32,33] can be modified to provide also the initial estimate of the viewpoint shift.

Let us assume that $(u_{n,j}, v_{n,j})$, $n = 1, \dots, N$, $j = 1, \dots, J$, are the checkerboard corner points detected in the captured calibration image series. In particular, the point $(u_{n,j}, v_{n,j})$ is the position of the point (X_n, Y_n) on the calibration chart in the j th calibration image. To initialize the parameters of the affine or projective transform representing the mapping from the ideal image plane to the image

pixel coordinates we approximate it by a translation shifting the origin of the coordinate system to the center of the image. Alternatively, a more accurate initial guess could be obtained by using one of existing methods for estimating the center of radial distortion (c_1, c_2) , such as [16, 17], or the initial affine transform could be estimated by using the circular boundary of the camera field of view, if it is available, as in [27]. However in our experiments we found that unnecessary.

Let us denote $(x_{n,j}, y_{n,j})$ the resulting estimated points in the ideal image plane corresponding to the detected image points $(u_{n,j}, v_{n,j})$. We want to find the coefficients of polynomials f and g in (4) and by (8), respectively, and the set of 3x3 rotation matrices \mathbf{R}_j and 3x1 translation vectors \mathbf{t}_j , $j = 1, \dots, J$, such that

$$\mathbf{R}_j \begin{bmatrix} X_n \\ Y_n \\ 0 \end{bmatrix} + \mathbf{t}_j = \begin{bmatrix} 0 \\ 0 \\ g(\rho_{n,j}) \end{bmatrix} + d_{n,j} \begin{bmatrix} x_{n,j} \\ y_{n,j} \\ f(\rho_{n,j}) \end{bmatrix} \quad (13)$$

for some $d_{n,j} > 0$. As usual, $\rho_{n,j} = \sqrt{x_{n,j}^2 + y_{n,j}^2}$.

We can follow the example set in [32, 33] and eliminate the unknown distance $d_{n,j}$ along the ray by using cross product. In our case, we have

$$\left(\mathbf{R}_j \begin{bmatrix} X_n \\ Y_n \\ 0 \end{bmatrix} + \mathbf{t}_j - \begin{bmatrix} 0 \\ 0 \\ g(\rho_{n,j}) \end{bmatrix} \right) \times \begin{bmatrix} x_{n,j} \\ y_{n,j} \\ f(\rho_{n,j}) \end{bmatrix} = \mathbf{0}. \quad (14)$$

If we denote

$$\mathbf{R}_j = \begin{bmatrix} r_{11}^{(j)} & r_{12}^{(j)} & r_{13}^{(j)} \\ r_{21}^{(j)} & r_{22}^{(j)} & r_{23}^{(j)} \\ r_{31}^{(j)} & r_{32}^{(j)} & r_{33}^{(j)} \end{bmatrix}, \quad \mathbf{t}_j = \begin{bmatrix} t_1^{(j)} \\ t_2^{(j)} \\ t_3^{(j)} \end{bmatrix}, \quad (15)$$

this can be re-written as

$$(r_{21}^{(j)} X_n + r_{22}^{(j)} Y_n + t_2^{(j)})f(\rho_{n,j}) - \quad (16)$$

$$(r_{31}^{(j)} X_n + r_{32}^{(j)} Y_n + t_3^{(j)} - g(\rho_{n,j}))y_{n,j} = 0$$

$$(r_{31}^{(j)} X_n + r_{32}^{(j)} Y_n + t_3^{(j)} - g(\rho_{n,j}))x_{n,j} - \quad (17)$$

$$(r_{11}^{(j)} X_n + r_{12}^{(j)} Y_n + t_1^{(j)})f(\rho_{n,j}) = 0$$

$$(r_{11}^{(j)} X_n + r_{12}^{(j)} Y_n + t_1^{(j)})y_{n,j} - \quad (18)$$

$$(r_{21}^{(j)} X_n + r_{22}^{(j)} Y_n + t_2^{(j)})x_{n,j} = 0$$

The last of the three equations, (18), does not involve the unknown polynomials f and g and is linear in the remaining unknowns. Collecting these equations for all the detected corners in a single calibration image, i.e., for all $n = 1, \dots, N$, we can form for each $j = 1, \dots, J$ a linear system that we can solve for $r_{11}^{(j)}$, $r_{12}^{(j)}$, $r_{21}^{(j)}$, $r_{22}^{(j)}$, $t_1^{(j)}$, and $t_2^{(j)}$. However, as this linear system is overdetermined and homogeneous, we can only find the solution in the least squares sense and it involves an unknown scale factor.

Let us denote this solution $\tilde{r}_{11}^{(j)}, \tilde{r}_{12}^{(j)}, \tilde{r}_{21}^{(j)}, \tilde{r}_{22}^{(j)}, \tilde{t}_1^{(j)}$, and $\tilde{t}_2^{(j)}$ and set

$$\mathbf{A} = \begin{bmatrix} \tilde{r}_{11}^{(j)} & \tilde{r}_{12}^{(j)} \\ \tilde{r}_{21}^{(j)} & \tilde{r}_{22}^{(j)} \end{bmatrix}. \quad (19)$$

To find the entries of the rotation matrix \mathbf{R}_j , we need to find 2x1 vectors \mathbf{b} , \mathbf{c} , and constants α and β such that the matrix

$$\begin{bmatrix} \alpha \mathbf{A} & \mathbf{b} \\ \mathbf{c}^T & \beta \end{bmatrix}. \quad (20)$$

is a rotation, i.e., it is orthogonal and its determinant is 1. There are four possible solutions to this completion problem. It is possible to show that the scaling factor α and vector \mathbf{c} must satisfy

$$\alpha = \pm \frac{1}{\sigma_1}, \quad \mathbf{c} = \pm \sqrt{1 - \frac{\sigma_2}{\sigma_1}} \mathbf{v}_2, \quad (21)$$

where σ_1 and σ_2 are the singular values of \mathbf{A} , $\sigma_1 \geq \sigma_2$, and \mathbf{v}_2 is the right singular vector corresponding to σ_2 . The last column of the completion can be found by computing the cross product of the first two columns, making sure that $\det(\mathbf{R}_j) = 1$. The elements of the translation vector are scaled by the same factor as the elements of the rotation matrix and thus

$$t_1^{(j)} = \alpha \tilde{t}_1^{(j)}, \quad t_2^{(j)} = \alpha \tilde{t}_2^{(j)}. \quad (22)$$

Caution needs to be exercised because only one of the four possible completions is the right rotation matrix \mathbf{R}_j . To find the correct signs, we check that the selected α and vector \mathbf{c} yield $d_{n,j} > 0$ in (13) and $f_0 > 0$ when the coefficients of f are estimated.

To estimate $t_3^{(j)}$ and the coefficients of f and g , we similarly as Scaramuzza create a single large system of linear equations for all n and j that needs to be solved. However, instead of using (16) and (17), as they suggest, we prefer to use the first two rows of (13) to estimate $d_{j,n}$ and then collect all the third rows involving the unknown coefficients,

$$\sum_{m=1}^M g_{2m} \rho_{n,j}^{2m} + d_{n,j} \sum_{k=0}^K f_{2k} \rho_{n,j}^{2k} - t_3^{(j)} = r_{31}^{(j)} X_n + r_{32}^{(j)} Y_n, \quad (23)$$

The resulting system of equations is smaller and, in our experiments, it also tends to provide more accurate results.

Note that the presented algorithm only requires that the spatial points corresponding to points detected in the calibration images lie in a single plane and their mutual position is known. It does not require the points to form a square grid. Hence, it allows both using a completely different type of a calibration chart, if desired, and it also makes

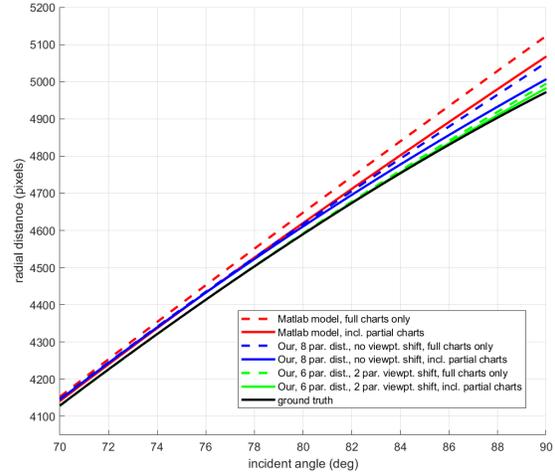


Figure 6. Synthetic data experiment using Nikkor 16mm f/2.8D design parameters, ground truth radial distortion curve and the radial distortion curves obtained by calibration with different models.

it possible to use images in which the chart is only partially visible. In that case, one simply collects only the equations corresponding to the corner points that are successfully detected in each image. We find this very useful, as it helps both to reduce the number of images that need to be captured for calibrating our circular array, and to improve the accuracy of calibration for the areas farther from the center. When only images in which the whole chart is visible can be used, the percentage of the corner points that lie farther out tends to be very small and thus significant error in the calibration on the periphery of the field of view can be easily outweighed by a minor improvement for the points closer to the center. The impact of including images in which the chart is only partially visible on the accuracy of calibration can be seen in Figure 6.

4.2. Non-linear optimization step

The parameters obtained by the algorithm described above generally do not provide sufficiently accurate characterization of the camera that could be used in practice and they only can serve as an initial guess for a subsequent non-linear refinement. Except for one case that is explicitly labeled, in all the practical experiments shown in Section 5 we used the most common cost function, the total square reprojection error, which was minimized by Levenberg-Marquardt algorithm (see, e.g., [15,23]). We implement the cost as suggested by Urban *et al.* in [37], making the error in horizontal and vertical direction two separate elements in the cost vector. Let $\Delta \mathbf{u}$ be the vector comprising elements $u'_{j,n} - u_{j,n}$, where $u_{j,n}$ is the column coordinate of the ac-

	reprojection error (pixels)		principal point (pixels)		distortion curve (pixels)		chart rotation (degrees)		chart translation (mm)	
	avg	max	avg	max	avg	max	avg	max	avg	max
Matlab model	1.45	36.98	1.38	3.04	12.07	110.56	0.21	5.90	1.84	32.82
Our 4-par. dist., affine, no vp. shift	1.00	20.78	1.40	2.74	10.11	81.78	0.16	3.28	1.77	20.73
Our 4-par. dist., no vp. shift	1.00	20.79	1.11	1.67	10.12	81.82	0.16	3.24	1.75	20.12
Our 4-par. dist., 2-par. vp. shift	0.69	22.15	0.38	0.72	3.41	59.94	0.08	2.33	0.46	13.38
Our 6-par. dist., no vp. shift	0.81	10.88	0.87	1.38	8.16	49.46	0.15	1.18	1.72	10.09
Our 6-par. dist., 2-par. vp. shift	0.31	2.76	0.12	0.29	0.59	20.56	0.01	0.19	0.14	1.08
Our 8-par. dist., no vp. shift	0.80	10.80	0.86	1.37	7.83	41.27	0.15	1.17	1.72	10.10

Table 1. Calibration reprojection error and calibrated model error with respect to ground truth for synthetic data based on Nikon D850 camera with 16mm f/2.8D lens.

	Calibration		Point prediction test error				
	Camera MRE (pixels)	Array MRE (pixels)	Max (pixels)	Average (pixels)	90th percentile (pixels)	Under 1 pixel (%)	Over 5 pixels (%)
Matlab model	0.50	0.83	69.95	2.80	6.41	36.71	10.98
Our 4-par. dist., affine, no vp. shift	0.40	0.74	65.03	2.49	5.91	42.40	9.49
Our 4-par. dist., no vp. shift	0.39	0.66	58.14	2.13	5.29	50.03	8.07
Our 4-par. dist., 2-par. vp. shift	0.30	0.38	16.14	0.93	2.17	71.22	1.33
Our 6-par. dist., no vp. shift	0.36	0.63	53.93	2.04	4.95	53.88	7.40
Our 6-par. dist., 2-par. vp. shift	0.25	0.31	11.71	0.52	1.20	86.57	0.17
Our 8-par. dist., no vp. shift	0.36	0.63	54.90	2.04	4.96	53.95	7.40
Our 4-par. d., 2-par. vp. shift, sp. cost	0.32	0.39	16.70	0.99	2.33	68.83	1.48

Table 2. Comparison of the accuracy of different models when calibrating real-life fisheye camera array

tual detected image point and $u'_{j,n}$ is the column coordinate of the image point that is the projection of the matching chart point (X_n, Y_n) , using the current estimates of the extrinsic rotation \mathbf{R}_j (represented by three parameters), translation \mathbf{t}_j , and all the intrinsic parameters. Analogously, let $\Delta \mathbf{v}$ be the vector comprising row coordinate differences. The cost that is minimized then can be expressed as

$$\Delta \mathbf{u}^T \Delta \mathbf{u} + \Delta \mathbf{v}^T \Delta \mathbf{v}. \quad (24)$$

As we have already mentioned in Section 4.1, we allow in our implementation variable number of points per image. So, in our case, vectors $\Delta \mathbf{u}$ and $\Delta \mathbf{v}$ collect the error only for all chart points that are visible and are successfully detected.

It is possible to use other cost functions. For example, to reduce the impact of outliers among the detected points, one can use instead l_1 cost or one of the robust costs proposed in literature. See e.g., [9, 36, 40]. In case of our model, an attractive choice can be a cost function that does not require projecting the chart points (X_n, Y_n) to the fisheye images, but it is instead based on mapping the detected image points to rays or points in 3D space. An example of such cost is

$$\Delta \mathbf{X}^T \Delta \mathbf{X} + \Delta \mathbf{Y}^T \Delta \mathbf{Y}, \quad (25)$$

where $\Delta \mathbf{X}$ and $\Delta \mathbf{Y}$ collect the differences between the actual chart corner points $(X_n, Y_n, 0)$ and their estimates $(X'_{n,j}, Y'_{n,j}, 0)$ produced with the current parameter values. As with the discussed models the mapping from 2D image to 3D space is much simpler, it makes the calibration significantly faster (about $10\times$ in our experiments).

The non-linear optimization step can take from a few seconds to few minutes. Its speed varies widely depending on the number of calibration images, stopping criteria of the optimization process, and other factors. When the total square reprojection error is used as the cost, it tends to be slightly slower than Matlab Computer Vision Toolbox, due to the larger number of intrinsic parameters that we use. However, we find the slower speed acceptable as, due to the need to capture and process many images of the chart, calibration with a chart is inherently a slow process suitable only for off-line use in situations when accuracy is more important than speed.

5. Experiment results

To compare different models we have run experiments with both synthetic and real captured data. We had to rely

mostly on our own data as the few publicly available sets such as those included with [10, 37] are not suitable for our purposes. The synthetic images for which ground truth is available are invariably produced with a central model and the available sets of real images are not rich enough to provide comparison beyond checking calibration mean reprojection error, which can be highly misleading. Not only it is strongly dependent on the point set used, but we also find that, for a single camera, errors in estimated intrinsics and extrinsics can partially offset each other and the resulting mean reprojection error can be deceptively low. See the results in Table 2.

To provide a fair comparison of different models, all the results shown are obtained with our own code that allows us to use exactly the same settings for all the models and to use images with only partially visible chart to make sure that the outer parts of the field of view are properly represented during calibration. However, to validate our code, we did compare the results it produces to those obtained with Matlab function `estimateFisheyeParameters`. When we use exactly the same point set and the same model as Matlab, the results tend to be virtually identical.

5.1. Synthetic data experiments

To produce synthetic data that is close enough to reality, we used the design specifications of Nikon D850 camera [3] with Nikkor 16mm f/2.8D fisheye lens. The lens characteristics – the radial distance of the projected point in the ideal image plane and the amount of the viewpoint shift for different incident angles – were obtained by ray tracing in Zemax, using the lens design data listed in [35], adopted from the U.S. patent 5,434,713. A randomly selected small offset was added to move the principal point off the center of the image. The chart positions used were generated randomly, but in order to keep the values realistic, the chart angle and distance distributions were based on those we observed in practice when calibrating our real 16-camera array. Table 1 lists the average and maximum error over 16 instances of the camera that we simulated, with respect to the ground truth.

Comparing the first two rows of Table 1 we see that replacing the distortion polynomial (1) with the polynomial (4) with the same number of parameters does help to improve the accuracy of the resulting calibration. Switching from affine to projective transform does not have any effect here because the sensor tilt was not included in this simulation, because of the lack of information about the realistic range of misalignment angles. The remaining five rows of Table 1 demonstrate that increasing the order of polynomial representing lens radial distortion can help to improve the accuracy some extent, but even better results are obtained when the viewpoint shift characterization is included.

You can see the impact of using different models on the

accuracy of the calibration also in Figure 6.

5.2. Real-life fisheye array calibration

With real-life data, in the absence of ground truth for the calibration, we use for evaluating the accuracy of calibration a common practical task the accuracy of which we can measure. The calibration image set is the output 360° fisheye camera array shown in Figure 1, comprising 16 Point Grey cameras equipped with Fujinon FE185C086HA-1 2.7 mm lenses. The captured images, see Figure 2, are of size 2448 × 2048 pixels and have approximately 185° field of view. We use image pairs captured by neighboring cameras to triangulate the spatial position of detected chart points and then measuring the error of predicting their location in the images captured by the remaining cameras.

Images capturing 41 different chart positions used as the test set provided us with 81,835 predicted locations for which we could reliably measure the distance from actual detected image points. The prediction error statistics summarized in Table 2 show quite clearly that both our proposed modifications of the central model, and incorporating the viewpoint shift in the model help to significantly increase the accuracy of calibration. Our best result provides an improvement by almost an order of magnitude. Note also that the accuracy of calibration with the spatial cost (25), which is considerably faster, is almost as good as the accuracy of calibration with the same model using the traditional reprojection error cost (24).

6. Conclusions

In this paper we propose modifications to the central fisheye model developed by Scaramuzza *et al.*, used by Matlab Computer Vision Toolbox and an extension of this model that incorporates the characterization of axial viewpoint shift exhibited by fisheye lenses. and we demonstrated that the techniques suggested in [32, 33] can be adapted to this non-central model. This allows switching between the simpler central projection and the more accurate non-central projection as needed, and also makes it possible to produce a user friendly tool for calibrating the non-central model that does not require lens design information that can be hard to obtain. Our experiments with both synthetic and real data show that the proposed modifications allow producing much more accurate fisheye camera calibration, which can help to significantly improve the performance of 3D computer vision and computational imaging systems involving fisheye cameras, particularly when objects close to the camera are present in the scene.

References

- [1] Fisheye calibration basics. Matlab documentation, <https://www.mathworks.com/help/vision/ug/fisheye-calibration-basics.html>. 1, 2

- [2] Fisheye camera model. OpenCV documentation, https://docs.opencv.org/3.4/db/d58/group__calib3d__fisheye.html. 2
- [3] Nikon D850 technical page. <https://www.nikonusa.com/en/nikon-products/d850-technical.page>. 8
- [4] Oleksandr Bogdan, Viktor Eckstein, François Rameau, and Jean-Charles Bazin. DeepCalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production (CVMP)*, pages 1–10, December 2018. 1
- [5] Pierre-Andre Brousseau and Sebastien Roy. Calibration of axial fisheye cameras through generic virtual central models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4039–4047, 2019. 3
- [6] Duane C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966. 3
- [7] Duane C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971. 3
- [8] Federico Camposco, Torsten Sattler, and Marc Pollefeys. Non-parametric structure-based calibration of radially symmetric cameras. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 3
- [9] Zhijun Dai, Fengjun Zhang, and Hongan Wang. Robust maximum likelihood estimation by sparse bundle adjustment using the L1 norm. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1672–1679, 2012. 7
- [10] Andrea Eichenseer and Andre Kaup. A data set providing synthetic and real-world fisheye video sequences. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1541–1545, 2016. 8
- [11] Peter Fasogbon and Emre Aksu. Calibration of fisheye camera using entrance pupil. In *IEEE International Conference on Image Processing (ICIP)*, pages 469–473, 2019. 3
- [12] Oliver Frank, Roman Katz, Christel-Loic Tisse, and Hugh Durrant-Whyte. Camera calibration for miniature, low-cost, wide-angle imaging systems. In *BMVC*, January 2007. 4
- [13] Donald Gennery. Generalized camera calibration including fish-eye lenses. *International Journal of Computer Vision*, 68:239–266, 2006. 3, 4
- [14] Nuno Gonçalves and Helder Araújo. Projection model, 3D reconstruction and rigid motion estimation from non-central catadioptric images. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 325 – 332, 2004. 3
- [15] Andrew Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2006. 4, 6
- [16] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1309–1321, 2007. 5
- [17] Fuyu Huang, Yongzhong Wang, Xueju Shen, Chao Lin, and Yudan Chen. Method for calibrating the fisheye distortion center. *Applied Optics*, 51(34):8169–8176, 2012. 5
- [18] Kenichi Kanatani. Calibration of ultra-wide fisheye lens cameras by eigenvalue minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):813–822, 2013. 1
- [19] Juho Kannala and Sami Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006. 1
- [20] Nobuyuki Kita. Evaluation of 3D analysis error caused by SVP approximation of fisheye lens. In *10th International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 357–368, 2015. 3
- [21] Nobuyuki Kita. Evaluation of the viewpoint shift for a fish-eye lens based on stereo geometry. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–5, 2016. 3
- [22] Avinash Kumar and Narendra Ahuja. On the equivalence of moving entrance pupil and radial distortion for camera calibration. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2345–2353, December 2015. 3, 4
- [23] Donald Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963. 6
- [24] Christopher Mei and Patrick Rives. Single view point omnidirectional camera calibration from planar grids. In *International Conference on Robotics and Automation*, pages 3945–3950, 2007. 1
- [25] Branislav Mičušík and Tomáš Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 1
- [26] Branislav Mičušík and Tomáš Pajdla. Autocalibration and 3D reconstruction with non-central catadioptric cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. 3
- [27] Branislav Mičušík and Tomáš Pajdla. Structure from motion with wide circular field of view cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1135–49, 2006. 5
- [28] Milan Šonka, Václav Hlaváč, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Cengage Learning, 4th edition, 2014. 4
- [29] Albert Pozo, Michael Toksvig, Terry Schrager, Joyce Hsu, Uday Mathur, Alexander Sorkine-Hornung, Rick Szeliski, and Brian Cabral. An integrated 6DoF video camera and system design. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38:1–16, 2019. 3
- [30] Luis Puig, Jesus Bermudez-Cameo, Peter Sturm, and Josechu Guerrero. Calibration of omnidirectional cameras in practice: A comparison of methods. *Computer Vision and Image Understanding*, 116(1):120–137, 2012. 1
- [31] Davide Scaramuzza. OCamCalib: Omnidirectional camera calibration toolbox for Matlab. <https://sites.google.com/site/scarobotix/ocamcalib-toolbox>. 2
- [32] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *IEEE International Conference on Computer Vision Systems (ICVS)*, 2006. 1, 2, 3, 5, 8

- [33] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, 2006. [2](#), [3](#), [4](#), [5](#), [8](#)
- [34] Rahul Swaminathan, Michael Grossberg, and Shree Nayar. Non-single viewpoint catadioptric cameras: Geometry and analysis. *International Journal of Computer Vision*, 66:211–229, 2006. [3](#)
- [35] Pierre Toscani. Fisheyes (in English). [http://www.pierretoscani.com/fisheyes-\(in-english\).html](http://www.pierretoscani.com/fisheyes-(in-english).html). [2](#), [8](#)
- [36] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science*, pages 198–372. Springer Verlag, 2000. [7](#)
- [37] Steffen Urban, Jens Leitloff, and Stefan Hinz. Improved wide-angle, fisheye and omnidirectional camera calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:72–79, 2015. [2](#), [3](#), [6](#), [8](#)
- [38] Vladyslav Usenko, Nikolaus Demmel, and Daniel Cremers. The double sphere camera model. In *International Conference on 3D Vision*, September 2018. [1](#)
- [39] Ye-Kui Wang, Miska M. Hannuksela, and Sachin Deshpande. Working draft 2 of ISO/IEC 23090-2, Information technology - Coded representation of immersive media - Part 2: Omnidirectional media format, July 2018. [2](#)
- [40] Christopher Zach. Robust bundle adjustment revisited. In *European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science*, volume 8693, pages 772–787, 2014. [7](#)