Calibration of Ultra-Wide Fisheye Lens Cameras by Eigenvalue Minimization

Ryohei Nakamura, Teppei Marumo and Kenichi Kanatani
Department of Computer Science, Okayama University, Japan
E-mail: {nakamura,marumo,kanatani}@suri.cs.okayama-u.ac.jp

Abstract. We present a new technique for calibrating an ultra-wide fisheye lens camera by imposing the constraint that collinear points be rectified to be collinear, parallel lines to be parallel, and orthogonal lines to be orthogonal. Exploiting the fact that line fitting reduces to an eigenvalue problem, we do a rigorous perturbation analysis to obtain a Levenberg-Marquardt procedure for the optimization. Doing experiments, we point out that spurious solutions exist if collinearity and parallelism alone are imposed. Our technique has many desirable properties. For example, no metric information is required about the reference pattern or the camera position, and separate stripe patterns can be displayed on a screen to generate a virtual grid, eliminating the grid point detection.

1. Introduction

Fisheye lens cameras are widely used for surveillance purposes because of their wide angles of view. They are also mounted on vehicles for various purposes including obstacle detection, self-localization, and bird's eye view generation [9, 12]. However, fisheye lens images have a large distortion, so that in order to apply the computer vision techniques accumulated for the past decades, one first needs to rectify the image into a perspective view. Already, there is a lot of literature for this [2, 4, 6, 7, 10, 11, 12, 13, 16, 17].

The standard technique is to place a reference grid plane and match the grid image with the reference grid, whose precise geometry is assumed to be known [2, 3, 4, 6, 17]. However, this approach is not very practical for recently popularized ultra-wide fisheye lenses, because they cover more than 180° angles of view and hence any (even infinite) reference plane cannot cover the entire field of view. This difficulty can be circumvented by using the *collinearity constraint* pointed out repeatedly, first by Onodera and Kanatani [14] in 1992, later by Swaminathan and Nayar [16] in 2000, and by Devernay and Faugeras [1] in 2001. They pointed out that camera calibration can be done by imposing the constraint that straight lines should be rectified to be straight. This constraint was used to calibrate fisheye lenses by Nakano, et al. [11], Kase et al. [7], and Okutsu et al. [12]. Komagata et al. [8] further introduced the *parallelism constraint* and the *orthogonality constraint*, requiring that parallel lines be rectified to be parallel and orthogonal lines to be orthogonal. However, the cost function has been directly minimized by brute force means such as the Brent method and the Powell method [15].

In this paper, we adopt the collinearity-parallelism-orthogonality constraint of Komagata et al. [8] and optimize it by *eigenvalue minimization*. The fact that imposing collinearity implies eigenvalue minimization and that the optimization can be done by invoking the perturbation theorem was pointed out by Onodera and Kanatani [14]. Using this principle, they rectified perspective images by gradient decent. Here, we consider ultra-wide fisheye lenses and do a more detailed perturbation analysis to derive the Levenberg-Marquardt (LM) procedure, currently regarded as the standard tool for efficient and accurate optimization.

We also show by experiments that the orthogonality constraint plays an essential role for ultra-wide fisheye lenses and that correct calibration cannot be done only by imposing collinearity or parallelism or both, pointing out the existence of a *spurious solution*. This fact has not been known in the past collinearity-based work [1, 7, 11, 12, 14, 16].

For data acquisition, we take images of stripe patterns of different orientations on a large-screen display by placing the camera in various positions. Like the past collinearity-based methods [1, 7, 11, 12, 14, 16], our method is *non-metric* in the sense that no metric information is required about the camera position or the reference pattern. Yet, many researchers pointed out the necessity of some auxiliary information. For example, Nakano et al. [10, 11] proposed vanishing point estimation using conic fitting to straight line images (recently, Hughes et al. [4] proposed this same technique again). Okutsu et al. [13] picked out the images of antipodal points by hand. Such auxiliary information may be useful to suppress spurious solutions. However, we show that accurate calibration is possible without any auxiliary information if we do eigenvalue minimization for the collinearity-parallelism-orthogonality constraint.

Section 2 describes our imaging geometry model. Section 3 gives derivative expressions of the fundamental quantities, followed by a detailed perturbation analysis of the collinearity constraint in Sec. 4, of the parallelism constraint in Sec. 5, and of the orthogonality constraint in Sec. 6. Section 7 describes our Levenberg-Marquardt (LM) optimization procedure. In Sec. 8, we show our experiments and an examples of real scene applications. In Sec. 9, we conclude.

2. Geometry of ultra-wide fisheye lens

We consider recently popularized ultra-wide fisheye lenses with the imaging geometry modeled by the

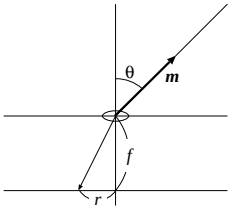


Figure 1 The imaging geometry of a fisheye lens and the incident ray vector $\boldsymbol{m}$.

stereographic projection

$$r = 2f \tan\frac{\theta}{2}, \tag{1}$$

where θ is the incidence angle (the angle of the incident ray of light from the optical axis) and r (in pixels) is the distance of the corresponding image point from the principal point (Fig. 1). The constant f is called the *focal length*. We consider Eq. (1) merely because our camera is as such, but the following calibration procedure is identical whatever model is used.

For a manufactured lens, the value of f is unknown or may not be exact if provided by the manufacturer. Also, the principal point may not be at the center of the image frame. Moreover, Eq. (1) is an idealization, and a real lens may not exactly satisfy it. So, we generalize Eq. (1) into the form

$$\frac{r}{f_0} + a_1\Big(\frac{r}{f_0}\Big)^3 + a_2\Big(\frac{r}{f_0}\Big)^5 + \cdots = \frac{2f}{f_0}\tan\frac{\theta}{2}, \tag{2}$$

and determine the values of f, a_1, a_2, ... along with the principal point position. Here, f_0 is a scale constant to keep the powers r^k within a reasonable numerical range (in our experiment, we used the value $f_0 = 150$ pixels). Since a sufficient number of correction terms could approximate any function, the right-hand side of Eq. (2) could be any function of θ, e.g., the perspective projection model $(f/f_0)\tan\theta$ or the equidistance projection model $(f/f_0)\theta$. We adopt the stereographic projection model of Eq. (1) merely for the sake of simple initialization for our camera.

In Eq. (2), even power terms do not exist, because the lens has circular symmetry and hence r is an odd function of θ. We assume that the azimuthal angle of the projection is equal to that of the incident ray. In the past, these two were often assumed to be slightly different, and geometric correction of the resulting "tangential distortion" was studied. Currently, the lens manufacturing technology is highly advanced so that the tangential distortion can be safely ignored. If not, we can simply include the tangential distortion terms in Eq. (2), and the subsequent calibration procedure remains unchanged.

In the literature, many authors have assumed the model in the form of $r = c_1\theta + c_2\theta^3 + c_3\theta^5 + \cdots$ [6, 7, 12, 11]. As we see shortly, the value of θ for a specified r is necessary in each step of the optimization iterations. Many authors computed θ by solving a polynomial equation by a numerical means [6, 7, 12, 11], but this causes loss of accuracy and efficiency. It is more convenient to assume the expansion of θ in terms of r from the beginning. From Eq. (2), we can express θ in terms of r in the form

$$\theta = 2\tan^{-1}\Big(\frac{f_0}{2f}\Big(\frac{r}{f_0} + a_1\Big(\frac{r}{f_0}\Big)^3 + a_2\Big(\frac{r}{f_0}\Big)^5 + \cdots\Big)\Big). \tag{3}$$

3. Incident ray vector and its derivatives

Let $\boldsymbol{m}$ be the unit vector in the direction of the incident ray of light which focuses at (x, y) on the image plane (Fig. 1). We call $\boldsymbol{m}$ the *incident ray vector*. In polar coordinates, it has the expression

$$\boldsymbol{m} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}, \tag{4}$$

where θ is the incidence angle from the Z-axis and ϕ is the azimuthal angle from the X-axis. If the principal point is at (u_0, v_0), we have

$$x - u_0 = r\cos\phi, \qquad y - v_0 = r\sin\phi,$$
$$r = \sqrt{(x-u_0)^2 + (y-v_0)^2}. \tag{5}$$

Hence, Eq. (4) is rewritten as

$$\boldsymbol{m} = \begin{pmatrix} ((x-u_0)/r)\sin\theta \\ ((y-v_0)/r)\sin\theta \\ \cos\theta \end{pmatrix}. \tag{6}$$

Differentiating this with respect to u_0 and v_0, we obtain

$$\frac{\partial\boldsymbol{m}}{\partial u_0} = -\frac{\sin\theta}{r}\boldsymbol{i}, \qquad \frac{\partial\boldsymbol{m}}{\partial v_0} = -\frac{\sin\theta}{r}\boldsymbol{j}, \tag{7}$$

where we define $\boldsymbol{i} \equiv (1,0,0)^\top$ and $\boldsymbol{j} \equiv (0,1,0)^\top$. Next, we consider derivation with respect to f. Differentiating Eq. (2) with respect to f on both sides, we have

$$0 = \frac{2}{f_0}\tan\frac{\theta}{2} + \frac{2f}{f_0}\frac{1}{2\cos^2(\theta/2)}\frac{\partial\theta}{\partial f}. \tag{8}$$

Hence, we obtain

$$\frac{\partial\theta}{\partial f} = -\Big(\frac{2}{f_0}\tan\frac{\theta}{2}\Big)\Big(\frac{f_0}{f}\cos^2\frac{\theta}{2}\Big)$$
$$= -\frac{2}{f}\sin\frac{\theta}{2}\cos\frac{\theta}{2} = -\frac{1}{f}\sin\theta. \tag{9}$$

If follows that the derivative of Eq. (6) with respect to f is

$$\frac{\partial\boldsymbol{m}}{\partial f} = \begin{pmatrix} ((x-u_0)/r)\cos\theta \\ ((y-v_0)/r)\cos\theta \\ -\sin\theta \end{pmatrix}\frac{\partial\theta}{\partial f}$$
$$= -\frac{1}{f}\sin\theta\begin{pmatrix} ((x-u_0)/r)\cos\theta \\ ((y-v_0)/r)\cos\theta \\ -\sin\theta \end{pmatrix}. \tag{10}$$

Finally, we consider derivation with respect to a_k. Differentiating (2) with respect to a_k on both sides, we have

$$\Big(\frac{r}{f_0}\Big)^{2k+1} = \frac{2f}{f_0}\frac{1}{2\cos^2(\theta/2)}\frac{\partial\theta}{\partial a_k}. \tag{11}$$

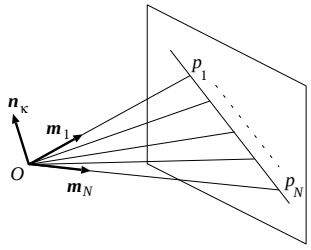


Figure 2 The incident ray vectors $\boldsymbol{m}_\alpha$ of collinear points p_1, ..., p_N are coplanar.

Hence, we obtain

$$\frac{\partial\theta}{\partial a_k} = \frac{f_0}{f}\Big(\frac{r}{f_0}\Big)^{2k+1}\cos^2\frac{\theta}{2}. \tag{12}$$

If follows that the derivative of Eq. (6) with respect to a_k is

$$\frac{\partial \boldsymbol{m}}{\partial a_k} = \begin{pmatrix} ((x-u_0)/r)\cos\theta \\ ((y-v_0)/r)\cos\theta \\ -\sin\theta \end{pmatrix}\frac{\partial\theta}{\partial a_k}$$
$$= \frac{f_0}{f}\Big(\frac{r}{f_0}\Big)^{2k+1}\cos^2\frac{\theta}{2}\begin{pmatrix} ((x-u_0)/r)\cos\theta \\ ((y-v_0)/r)\cos\theta \\ -\sin\theta \end{pmatrix}. \tag{13}$$

All the cost functions in the subsequent optimization are expressed in terms of the incident ray vector $\boldsymbol{m}$. Hence, we can compute derivatives of any cost function with respect to any parameter simply by combining the above derivative expressions,

4. Collinearity constraint

Suppose we observe a collinear point sequence $\mathcal{S}_\kappa$ (the subscript κ enumerates all existing sequences) consisting of N points p_1, ..., p_N, and let $\boldsymbol{m}_1$,..., $\boldsymbol{m}_N$ be their incident ray vectors. If the camera is precisely calibrated, the computed incident ray vectors should be coplanar (Fig. 2) and satisfy $(\boldsymbol{n}_\kappa, \boldsymbol{m}_\alpha) = 0$, $\alpha = 1$, ..., N. In the following, we denote the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $(\boldsymbol{a}, \boldsymbol{b})$. If the calibration is incomplete, $(\boldsymbol{n}_\kappa, \boldsymbol{m}_\alpha)$ may not be strictly zero. So, we adjust the parameters by minimizing

$$\sum_{\alpha\in\mathcal{S}_\kappa}(\boldsymbol{n}_\kappa, \boldsymbol{m}_\alpha)^2 = \sum_{\alpha\in\mathcal{S}_\kappa}\boldsymbol{n}_\kappa^\top\boldsymbol{m}_\alpha\boldsymbol{m}_\alpha^\top\boldsymbol{n}_\kappa$$
$$= (\boldsymbol{n}_\kappa, \sum_{\alpha\in\mathcal{S}_\kappa}\boldsymbol{m}_\alpha\boldsymbol{m}_\alpha^\top\boldsymbol{n}_\kappa) = (\boldsymbol{n}_\kappa, \boldsymbol{M}^{(\kappa)}\boldsymbol{n}_\kappa), \tag{14}$$

where we define

$$\boldsymbol{M}^{(\kappa)} = \sum_{\alpha\in\mathcal{S}_\kappa}\boldsymbol{m}_\alpha\boldsymbol{m}_\alpha^\top. \tag{15}$$

Equation (14) is a quadratic form of $\boldsymbol{M}^{(\kappa)}$, so its minimimum equals the smallest eigenvalue $\lambda_{\min}^{(\kappa)}$ of $\boldsymbol{M}^{(\kappa)}$. To enforce the collinearity constraint for all collinear point squences $\mathcal{S}_\kappa$, we determine the paremeters so as to minimize

$$J_1 = \sum_{\text{all }\kappa}\lambda_{\min}^{(\kappa)}. \tag{16}$$

4.1 First derivatives

We consider first derivatives of $\lambda_{\min}^{(\kappa)}$ with respect to c, which represnts the calibration parameters u_0, v_0, f, a_1, a_2, ... Differentiating the defining equation

$$\boldsymbol{M}^{(\kappa)}\boldsymbol{n}_\kappa = \lambda_{\min}^{(\kappa)}\boldsymbol{n}_\kappa \tag{17}$$

with respect to c on both sides, we have

$$\frac{\partial\boldsymbol{M}^{(\kappa)}}{\partial c}\boldsymbol{n}_\kappa + \boldsymbol{M}^{(\kappa)}\frac{\partial\boldsymbol{n}_\kappa}{\partial c} = \frac{\partial\lambda_{\min}^{(\kappa)}}{\partial c}\boldsymbol{n}_\kappa + \lambda_{\min}^{(\kappa)}\frac{\partial\boldsymbol{n}_\kappa}{\partial c}. \tag{18}$$

Computing the inner product with $\boldsymbol{n}_\kappa$ on both sides, we obtain

$$(\boldsymbol{n}_\kappa, \frac{\partial\boldsymbol{M}^{(\kappa)}}{\partial c}\boldsymbol{n}_\kappa) + (\boldsymbol{n}_\kappa, \boldsymbol{M}^{(\kappa)}\frac{\partial\boldsymbol{n}_\kappa}{\partial c})$$
$$= \frac{\partial\lambda_{\min}^{(\kappa)}}{\partial c}(\boldsymbol{n}_\kappa, \boldsymbol{n}_\kappa) + \lambda_{\min}^{(\kappa)}(\boldsymbol{n}_\kappa, \frac{\partial\boldsymbol{n}_\kappa}{\partial c}). \tag{19}$$

Since $\boldsymbol{n}_\kappa$ is a unit vector, we have $(\boldsymbol{n}_\kappa, \boldsymbol{n}_\kappa) = \|\boldsymbol{n}_\kappa\|^2 = 1$. Variations of a unit vector should be orthogonal to itself, so $(\boldsymbol{n}_\kappa, \partial\boldsymbol{n}_\kappa/\partial c) = 0$. Since $\boldsymbol{M}^{(\kappa)}$ is a symmetric matrix, we have $(\boldsymbol{n}_\kappa, \boldsymbol{M}^{(\kappa)}\partial\boldsymbol{n}_\kappa/\partial c) = (\boldsymbol{M}^{(\kappa)}\boldsymbol{n}_\kappa, \partial\boldsymbol{n}_\kappa/\partial c) = \lambda_{\min}^{(\kappa)}(\boldsymbol{n}_\kappa, \partial\boldsymbol{n}_\kappa/\partial c) = 0$. Thus, Eq. (19) implies

$$\frac{\partial\lambda_{\min}^{(\kappa)}}{\partial c} = (\boldsymbol{n}_\kappa, \frac{\partial\boldsymbol{M}^{(\kappa)}}{\partial c}\boldsymbol{n}_\kappa). \tag{20}$$

This result is well known as the *perturabation theorem* of eigenvalue problems [5]. From the definiton of $\boldsymbol{M}^{(\kappa)}$ in Eq. (45), we see that

$$\frac{\partial\boldsymbol{M}^{(\kappa)}}{\partial c} = \sum_{\alpha=1}^{N}\Big(\frac{\partial\boldsymbol{m}_\alpha}{\partial c}\boldsymbol{m}_\alpha^\top + \boldsymbol{m}_\alpha\Big(\frac{\partial\boldsymbol{m}_\alpha}{\partial c}\Big)^\top\Big)$$
$$= 2\mathcal{S}[\sum_{\alpha=1}^{N}\frac{\partial\boldsymbol{m}_\alpha}{\partial c}\boldsymbol{m}_\alpha^\top] \equiv \boldsymbol{M}_c^{(\kappa)}, \tag{21}$$

where $\mathcal{S}[\,\cdot\,]$ denotes symmetrization ($\mathcal{S}[\boldsymbol{A}] = (\boldsymbol{A} + \boldsymbol{A}^\top)/2$). Thus, the first derivatives of the function J_1 with respect to $c = u_0$, v_0, f, a_1, a_2, ... are given as follows:

$$\frac{\partial J_1}{\partial c} = \sum_{\text{all }\kappa}(\boldsymbol{n}_\kappa, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa). \tag{22}$$

4.2 Second derivatives

Differentiating Eq. (20) with respect to c' ($= u_0$, v_0, f, a_1, a_2, ...), we obtain

$$\frac{\partial^2\lambda_{\min}^{(\kappa)}}{\partial c\partial c'} = (\frac{\partial\boldsymbol{n}_\kappa}{\partial c'}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa) + (\boldsymbol{n}_\kappa, \frac{\partial^2\boldsymbol{M}^{(\kappa)}}{\partial c\partial c'}\boldsymbol{n}_\kappa)$$
$$+(\boldsymbol{n}_\kappa, \boldsymbol{M}_c^{(\kappa)}\frac{\partial\boldsymbol{n}_\kappa}{\partial c'})$$
$$= (\boldsymbol{n}_\kappa, \frac{\partial^2\boldsymbol{M}^{(\kappa)}}{\partial c\partial c'}\boldsymbol{n}_\kappa) + 2(\frac{\partial\boldsymbol{n}_\kappa}{\partial c'}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa). \tag{23}$$

First, consider the first term. Differentiation of Eq. (21) with respect to c' is

$$\frac{\partial^2\boldsymbol{M}^{(\kappa)}}{\partial c\partial c'} = 2\mathcal{S}[\sum_{\alpha=1}^{N}\Big(\frac{\partial^2\boldsymbol{m}_\alpha}{\partial c\partial c'}\boldsymbol{m}_\alpha^\top + \frac{\partial\boldsymbol{m}_\alpha}{\partial c}\Big(\frac{\partial\boldsymbol{m}_\alpha}{\partial c'}\Big)^\top\Big)], \tag{24}$$

and hence we have

$$(\boldsymbol{n}_\kappa, \frac{\partial^2 \boldsymbol{M}^{(\kappa)}}{\partial c \partial c'}\boldsymbol{n}_\kappa) = 2\sum_{\alpha=1}^{N}\Big((\boldsymbol{n}_\kappa, \frac{\partial^2 \boldsymbol{m}_\alpha}{\partial c \partial c'})(\boldsymbol{m}_\alpha, \boldsymbol{n}_\kappa) + (\boldsymbol{n}_\kappa, \frac{\partial \boldsymbol{m}_\alpha}{\partial c})(\frac{\partial \boldsymbol{m}_\alpha}{\partial c'}, \boldsymbol{n}_\kappa)\Big). \quad (25)$$

If the calibration is complete, we should have $(\boldsymbol{m}_\alpha, \boldsymbol{n}_\kappa) = 0$. In the course of the optimization, we can expect that $(\boldsymbol{m}_\alpha, \boldsymbol{n}_\kappa) \approx 0$. Hence, Eq. (25) can be approximated by

$$(\boldsymbol{n}_\kappa, \frac{\partial^2 \boldsymbol{M}^{(\kappa)}}{\partial c \partial c'}\boldsymbol{n}_\kappa) \approx 2\sum_{\alpha=1}^{N}(\boldsymbol{n}_\kappa, \frac{\partial \boldsymbol{m}_\alpha}{\partial c})(\frac{\partial \boldsymbol{m}_\alpha}{\partial c'}, \boldsymbol{n}_\kappa) = 2(\boldsymbol{n}_\kappa, \boldsymbol{M}_{cc'}^{(\kappa)}\boldsymbol{n}_\kappa), \quad (26)$$

$$\boldsymbol{M}_{cc'}^{(\kappa)} \equiv \sum_{\alpha=1}^{N}\Big(\frac{\partial \boldsymbol{m}_\alpha}{\partial c}\Big)\Big(\frac{\partial \boldsymbol{m}_\alpha}{\partial c'}\Big)^\top. \quad (27)$$

This is a sort of the Gauss-Newton approximation.

Next, consider the second term of Eq. (23). Since $\boldsymbol{n}_\kappa$ is a unit vector, its variations are orthogonal to itself. Let $\lambda_1^{(\kappa)} \geq \lambda_2^{(\kappa)} \geq \lambda_{\min}^{(\kappa)}$ be the eigenvalues of $\boldsymbol{M}^{(\kappa)}$ with $\boldsymbol{n}_{\kappa 1}$, $\boldsymbol{n}_{\kappa 2}$, and $\boldsymbol{n}_\kappa$ the corresponding unit eigenvectors. Since the eigenvectors of a symmetric matrix are mutually orthogonal, any vector orthogonal to $\boldsymbol{n}_\kappa$ is expressed as a linear combination of $\boldsymbol{n}_{\kappa 1}$ and $\boldsymbol{n}_{\kappa 2}$. Hence, we can write

$$\frac{\partial \boldsymbol{n}_\kappa}{\partial c} = \beta_1 \boldsymbol{n}_{\kappa 1} + \beta_2 \boldsymbol{n}_{\kappa 2}, \quad (28)$$

for some β_1 and β_2. Substitution of Eqs. (20) and (28) into Eq. (18) results in

$$\boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa + \boldsymbol{M}^{(\kappa)}(\beta_1 \boldsymbol{n}_{\kappa 1} + \beta_2 \boldsymbol{n}_{\kappa 2}) = (\boldsymbol{n}_\kappa, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)\boldsymbol{n}_\kappa + \lambda_{\min}^{(\kappa)}(\beta_1 \boldsymbol{n}_{\kappa 1} + \beta_2 \boldsymbol{n}_{\kappa 2}). \quad (29)$$

Noting that $\boldsymbol{M}^{(\kappa)}\boldsymbol{n}_{\kappa 1} = \lambda_1^{(\kappa)}\boldsymbol{n}_{\kappa 1}$ and $\boldsymbol{M}^{(\kappa)}\boldsymbol{n}_{\kappa 2} = \lambda_2^{(\kappa)}\boldsymbol{n}_{\kappa 2}$, we have

$$\beta_1(\lambda_1^{(\kappa)} - \lambda_{\min}^{(\kappa)})\boldsymbol{n}_{\kappa 1} + \beta_2(\lambda_2^{(\kappa)} - \lambda_{\min}^{(\kappa)})\boldsymbol{n}_{\kappa 2} = (\boldsymbol{n}_\kappa, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)\boldsymbol{n}_\kappa - \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa. \quad (30)$$

Computing the inner product with $\boldsymbol{n}_{\kappa 1}$ and $\boldsymbol{n}_{\kappa 2}$ on both sides, we obtain

$$\beta_1(\lambda_1^{(\kappa)} - \lambda_{\min}^{(\kappa)}) = -(\boldsymbol{n}_{\kappa 1}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa), \quad \beta_2(\lambda_2^{(\kappa)} - \lambda_{\min}^{(\kappa)}) = -(\boldsymbol{n}_{\kappa 2}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa). \quad (31)$$

Thus, Eq. (28) is written as follows:

$$\frac{\partial \boldsymbol{n}_\kappa}{\partial c} = -\frac{(\boldsymbol{n}_{\kappa 1}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)\boldsymbol{n}_{\kappa 1}}{\lambda_1^{(\kappa)} - \lambda_{\min}^{(\kappa)}} - \frac{(\boldsymbol{n}_{\kappa 2}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)\boldsymbol{n}_{\kappa 2}}{\lambda_2^{(\kappa)} - \lambda_{\min}^{(\kappa)}}. \quad (32)$$

This is also a well known result of the perturbation theorem of eigenvalue problems [5]. Thus, the second term of Eq. (23) can be written as

$$2(\frac{\partial \boldsymbol{n}_\kappa}{\partial c'}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa) = -\frac{2(\boldsymbol{n}_{\kappa 1}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)(\boldsymbol{n}_{\kappa 1}, \boldsymbol{M}_{c'}^{(\kappa)}\boldsymbol{n}_\kappa)}{\lambda_1^{(\kappa)} - \lambda_{\min}^{(\kappa)}} - \frac{2(\boldsymbol{n}_{\kappa 2}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)(\boldsymbol{n}_{\kappa 2}, \boldsymbol{M}_{c'}^{(\kappa)}\boldsymbol{n}_\kappa)}{\lambda_2^{(\kappa)} - \lambda_{\min}^{(\kappa)}}. \quad (33)$$

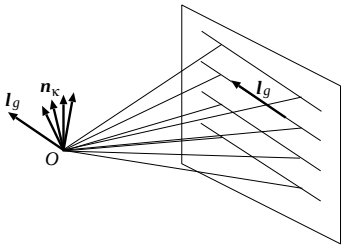


Figure 3 The surface normals $\boldsymbol{n}_\kappa$ to the planes defined by parallel lines are orthogonal to the common direction $\boldsymbol{l}_g$ of the lines.

Combining Eq. (26) and (33), we can approximate Eq. (23) in the form

$$\frac{\partial^2 \lambda_{\min}^{(\kappa)}}{\partial c \partial c'} \approx 2\Big((\boldsymbol{n}_\kappa, \boldsymbol{M}_{cc'}^{(\kappa)}\boldsymbol{n}_\kappa) - \sum_{i=1}^{2}\frac{(\boldsymbol{n}_{\kappa i}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)(\boldsymbol{n}_{\kappa i}, \boldsymbol{M}_{c'}^{(\kappa)}\boldsymbol{n}_\kappa)}{\lambda_i^{(\kappa)} - \lambda_{\min}^{(\kappa)}}\Big). \quad (34)$$

Thus, the second derivatives of the function J_1 with respect to c and c' are given by

$$\frac{\partial^2 J_1}{\partial c \partial c'} = 2\sum_{\text{all } \kappa}\Big((\boldsymbol{n}_\kappa, \boldsymbol{M}_{cc'}^{(\kappa)}\boldsymbol{n}_\kappa) - \sum_{i=1}^{2}\frac{(\boldsymbol{n}_{\kappa i}, \boldsymbol{M}_c^{(\kappa)}\boldsymbol{n}_\kappa)(\boldsymbol{n}_{\kappa i}, \boldsymbol{M}_{c'}^{(\kappa)}\boldsymbol{n}_\kappa)}{\lambda_i^{(\kappa)} - \lambda_{\min}^{(\kappa)}}\Big). \quad (35)$$

5. Parallelism constraint

Let $\mathcal{G}_g$ be a group of parallel collinear point sequences (the subscript g enumerates all existing groups) with a common orientation $\boldsymbol{l}_g$ (unit vector). The normals $\boldsymbol{n}_\kappa$ to the planes passing through the origin O (lens center) and lines of $\mathcal{G}_g$ are all orthogonal to $\boldsymbol{l}_g$ (Fig. 3). Hence, we should have $(\boldsymbol{l}_g, \boldsymbol{n}_\kappa) = 0$, $\kappa \in \mathcal{G}_g$, if the calibration is complete. So, we adjust the parameters by minimizing

$$\sum_{\kappa \in \mathcal{G}_g}(\boldsymbol{l}_g, \boldsymbol{n}_\kappa)^2 = \sum_{\kappa \in \mathcal{G}_g}\boldsymbol{l}_g^\top \boldsymbol{n}_\kappa \boldsymbol{n}_\kappa^\top \boldsymbol{l}_g = (\boldsymbol{l}_g, \sum_{\kappa \in \mathcal{G}_g}\boldsymbol{n}_\kappa \boldsymbol{n}_\kappa^\top \boldsymbol{l}_g) = (\boldsymbol{l}_g, \boldsymbol{N}^{(g)}\boldsymbol{l}_g), \quad (36)$$

where we define

$$\boldsymbol{N}^{(g)} = \sum_{\kappa \in \mathcal{G}_g}\boldsymbol{n}_\kappa \boldsymbol{n}_\kappa^\top. \quad (37)$$

Equation (36) is a quadratic form of $\boldsymbol{N}^{(g)}$, so its minimum equals the smallest eigenvalue $\mu_{\min}^{(g)}$ of $\boldsymbol{N}^{(g)}$. To enforce the parallelism constraint for all groups of parallel collinear sequences, we determine the parameters so as to minimize

$$J_2 = \sum_{\text{all } g}\mu_{\min}^{(g)}. \quad (38)$$

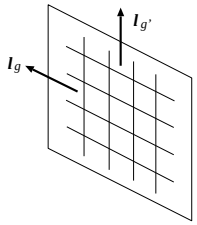


Figure 4 If two sets of parallel lines make right angles, their directions $\boldsymbol{l}_g$ and $\boldsymbol{l}_{g'}$ are orthogonal to each other.

5.1 First derivatives

Doing the same perturbation analysis as in Sec. 4, we obtain the first derivatives of the function J_2 with respect to parameters c in the form

$$\frac{\partial J_2}{\partial c} = \sum_{\text{all } g} (\boldsymbol{l}_g, \boldsymbol{N}_c^{(g)} \boldsymbol{l}_g),$$

$$\boldsymbol{N}_c^{(g)} = 2\mathcal{S}[\sum_{\kappa \in \mathcal{G}_g} \frac{\partial \boldsymbol{n}_\kappa}{\partial c} \boldsymbol{n}_\kappa^\top], \tag{39}$$

where $\boldsymbol{l}_g$ is the unit eigenvector of the matrix $\boldsymbol{N}^{(g)}$ in Eq. (37) for the smallest eigenvalue $\mu_{\min}^{(g)}$. The first derivative of $\partial \boldsymbol{n}_\kappa / \partial c$ is given by Eq. (32).

5.2 Second derivatives

Doing the same perturbation analysis as in Sec. 4, we obtain the second derivatives of the function J_2 with respect to parameters c and c' in the form

$$\frac{\partial^2 J_2}{\partial c \partial c'} = 2 \sum_{\text{all } g} \Big((\boldsymbol{l}_g, \boldsymbol{N}_{cc'}^{(g)} \boldsymbol{l}_g) - \sum_{i=1}^{2} \frac{(\boldsymbol{l}_{gi}, \boldsymbol{N}_c^{(g)} \boldsymbol{l}_g)(\boldsymbol{l}_{gi}, \boldsymbol{N}_{c'}^{(g)} \boldsymbol{l}_g)}{\mu_i^{(g)} - \mu_{\min}^{(g)}} \Big), \tag{40}$$

where $\mu_i^{(g)}$, $i = 1, 2$, are the first and the second largest eigenvalues of the matrix $\boldsymbol{N}^{(g)}$ and $\boldsymbol{l}_{gi}$ are the corresponding unit eigenvectors. The matrix $\boldsymbol{N}_{cc'}^{(g)}$ is defined by

$$\boldsymbol{N}_{cc'}^{(g)} \equiv \sum_{\kappa \in \mathcal{G}_g} \Big(\frac{\partial \boldsymbol{n}_\kappa}{\partial c} \Big) \Big(\frac{\partial \boldsymbol{n}_\kappa}{\partial c'} \Big)^\top. \tag{41}$$

6. Orthogonality constraint

Suppose we observe two groups $\mathcal{G}_g$ and $\mathcal{G}_{g'}$ of parallel line sequences with mutually orthogonal directions $\boldsymbol{l}_g$ and $\boldsymbol{l}_{g'}$ (Fig. 4). The orientation $\boldsymbol{l}_g$ of the sequences in the group $\mathcal{G}_g$ is the unit eigenvector of the matrix $\boldsymbol{N}^{(g)}$ in Eq. (37) for the smallest eigenvalue. If the calibration is complete, we should have $(\boldsymbol{l}_g, \boldsymbol{l}_{g'}) = 0$, so we adjust the parameters by minimizing

$$J_3 = \sum_{\substack{\text{all orthogonal} \\ \text{pairs } \{\mathcal{G}_g, \mathcal{G}'_g\}}} (\boldsymbol{l}_g, \boldsymbol{l}_{g'})^2. \tag{42}$$

6.1 First derivatives

The first derivatives of the function J_3 with respect to parameters c are given by

$$\frac{\partial J_3}{\partial c} = 2 \sum_{\substack{\text{all orthogonal} \\ \text{pairs } \{\mathcal{G}_g, \mathcal{G}'_g\}}} (\boldsymbol{l}_g, \boldsymbol{l}_{g'}) \Big((\frac{\partial \boldsymbol{l}_g}{\partial c}, \boldsymbol{l}_{g'}) + (\boldsymbol{l}_g, \frac{\partial \boldsymbol{l}_{g'}}{\partial c}) \Big). \tag{43}$$

The first derivative $\partial \boldsymbol{l}_g / \partial c$ is given by

$$\frac{\partial \boldsymbol{l}_g}{\partial c} = -\sum_{i=1}^{2} \frac{(\boldsymbol{l}_{gi}, \boldsymbol{N}_c^{(g)} \boldsymbol{l}_g) \boldsymbol{l}_{gi}}{\mu_i^{(g)} - \mu_{\min}^{(g)}}, \tag{44}$$

and $\partial \boldsymbol{l}_{g'} / \partial c$ similarly.

6.2 Second derivatives

Using the Gauss-Newton approximation $(\boldsymbol{l}_g, \boldsymbol{l}_{g'}) \approx 0$, we obtain the second derivatives of the function J_3 with respect to parameters c and c' in the form

$$\frac{\partial^2 J_3}{\partial c \partial c'} = 2 \sum_{\substack{\text{all orthogonal} \\ \text{pairs } \{\mathcal{G}_g, \mathcal{G}'_g\}}} \Big((\frac{\partial \boldsymbol{l}_g}{\partial c}, \boldsymbol{l}_{g'}) + (\boldsymbol{l}_g, \frac{\partial \boldsymbol{l}_{g'}}{\partial c}) \Big) \Big((\frac{\partial \boldsymbol{l}_g}{\partial c'}, \boldsymbol{l}_{g'}) + (\boldsymbol{l}_g, \frac{\partial \boldsymbol{l}_{g'}}{\partial c'}) \Big). \tag{45}$$

7. Levenberg-Marquardt procedure

To incorporate all of the collinearity, parallelism, and orthogonality constraints, we minimize

$$J = \frac{J_1}{\gamma_1} + \frac{J_2}{\gamma_1} + \frac{J_3}{\gamma_1}, \tag{46}$$

where γ_i, $i = 1, 2, 3$, are the weights to balance the magnitudes of the three terms. Note that $J_1 \gg J_2 \gg J_3$, since J_1 is proportional to the number of all points, J_2 to the number of all lines, and J_3 to the number of orthogonal pairs of parallel lines. In our experiment, we used as γ_i the initial value of J_i.

Since we have derived the first and second derivatives of all J_i with respect to all the parameters, we can now combine them into the following Levenberg-Marquardt (LM) procedure [15]:

1. Provide initial values, e.g., let the principal point (u_0, v_0) be at the frame center, f be a default value and, let $a_1 = a_2 = \cdots = 0$. Let J_0 be the value of the function J for these initial values, and let $C = 0.0001$.
2. Compute the incidence angle $\theta_{\kappa\alpha}$ of the αth point p_α in the κth sequence $\mathcal{S}_\kappa$ by Eq. (3), and compute its incidence ray vector $\boldsymbol{m}_{\kappa\alpha}$ by Eq. (6) Then, compute the derivatives $\partial \boldsymbol{m}_{\kappa\alpha} / \partial c$ by Eqs. (7), (10), and (13), $c = u_0$, v_0, f, a_1, a_2, ...
3. Compute the first derivatives J_c and the second derivatives $J_{cc'}$ of the function J, c, $c' = u_0$, v_0, f, a_1, a_2, ...

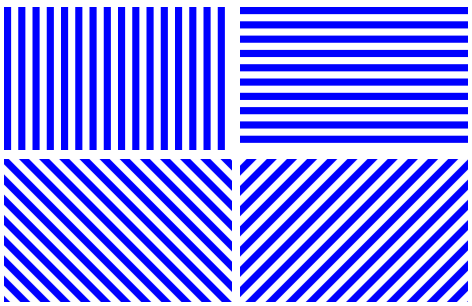
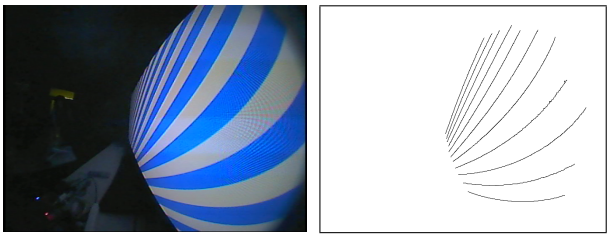


Figure 5 Stripe patterns in four directions.



(a) (b)

Figure 6 (a) Fisheye lens image of a stripe pattern. (b) Detected edges.

4. Determine the increments Δu_0, Δv_0, Δf, Δa_1, ... by solving the linear equation

$$\begin{pmatrix} (1+C)J_{u_0u_0} & J_{u_0v_0} & J_{u_0f} & J_{u_0a_1} & \cdots \\ J_{v_0u_0} & (1+C)J_{v_0v_0} & J_{v_0f} & J_{v_0a_1} & \cdots \\ J_{fu_0} & J_{fv_0} & (1+C)J_{ff} & J_{fa_1} & \cdots \\ J_{a_1u_0} & J_{a_1v_0} & J_{a_1f} & (1+C)J_{a_1a_1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \Delta u_0 \\ \Delta v_0 \\ \Delta f \\ \Delta a_1 \\ \vdots \end{pmatrix} = - \begin{pmatrix} J_{u_0} \\ J_{v_0} \\ J_f \\ J_{a_1} \\ \vdots \end{pmatrix}. \tag{47}$$

5. Tentatively update the parameter values in the form

$$\tilde{u}_0 = u_0 + \Delta u_0, \quad \tilde{v} = v_0 + \Delta v_0, \quad \tilde{f} = f + \Delta f,$$

$$\tilde{a}_1 = a_1 + \Delta a_1, \quad \tilde{a}_2 = a_2 + \Delta a_2, \quad ... \tag{48}$$

and evaluate the resulting value $\tilde{J}$ of the function J.

6. If $\tilde{J} < J_0$, proceed. Else, let $C \leftarrow 10C$ and go back to Step 4.
7. Let $u_0 \leftarrow \tilde{u}_0$, $v_0 \leftarrow \tilde{v}_0$, $f \leftarrow \tilde{f}$, $a_1 \leftarrow \tilde{a}_1$, $a_2 \leftarrow \tilde{a}_2$, ... If $|\Delta u_0| < \epsilon_0$, $|\Delta v_0| < \epsilon_0$, $|\Delta f| < \epsilon_f$, $|\Delta a_1| < \epsilon_1$, $|\Delta a_2| < \epsilon_2$, ..., return J, u_0, v_0, f, a_1, a_2, ..., and stop. Else, let $J_0 \leftarrow J$ and $C \leftarrow C/10$, and go back to Step 2.

8. Experiments

Figure 5 shows the four stripe patterns we used in our experiments. We displayed them on a large video screen and take their images by placing the camera in various positions so that the pattern image appears in various parts of the view (recall that the view cannot be covered by a single planar pattern image). The four patterns were repeatedly displayed cyclically with blank frames in-between, and the camera is fixed in each position at least for one round of the display to take images of the four types. Figure 6(a) is one shot of such images. The image size is 640×480 pixels. From each image, we detected edges; Fig. 6(b) shows the edges detected from the image in Fig. 6(a). We manually removed edges outside the display area. We also removed too small clusters of edge points. Then, we ran an edge segmentation algorithm to classify the remaining edge points into connected edge segments. On each segment was imposed the collinearity constraint; on the segments resulting from one stripe pattern were imposed the parallelism constraint; on the segments resulting from consecutive stripe patterns for a fixed camera position were imposed the orthogonality constraint. In all, we obtained 220 segments, consisting of 20 groups of parallel segments and 10 orthogonal pairs, to which the LM procedure was applied.

Let us call the number K of the terms on the left hand side of Eq. (2) the *correction degree*, meaning that the left-hand side of Eq. (2) is approximated by a $(2K+1)$th degree polynomial. The results up to the fifth correction degree are shown in Table 1. We set the frame center to be $(0, 0)$ to specify the principal point (u_0, v_0). For the convergence thresholds in Step 7 of the LM iterations, we let $\epsilon_0 = \epsilon_f = 10^{-3}$, $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-6}$, $\epsilon_3 = 10^{-7}$, $\epsilon_4 = 10^{-8}$, and $\epsilon_5 = 10^{-9}$. Using various different initial values, we confirmed that the LM always converges to the same solution after at most 10 iterations.

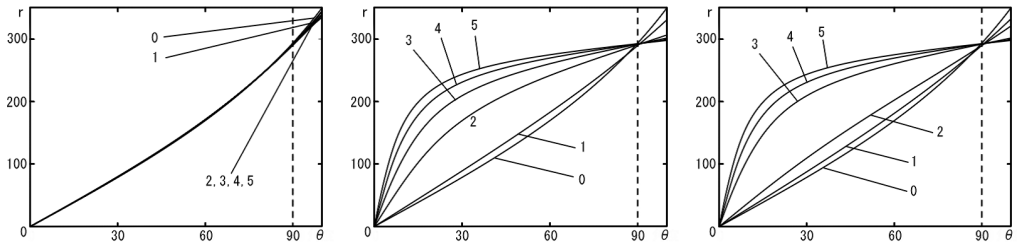
Figure 7(a) plots the graph of Eq. (3) for different correction degrees. For the convenience of applications, we numerically converted Eq. (3) to express the angle θ in terms of the distance r. As we see, the stereographic projection model in Eq. (1) holds fairly well even without any correction terms (degree 0). The result is almost unchanged for the degrees 3, 4, and 5 (i.e., including powers up to r^7, r^9, and r^{11}). Thus, there is no need to increase the correction terms any further.

For comparison, Fig. 7(b) shows the same result using collinearity alone; Fig. 7(c) shows the result collinearity and parallelism. In both cases, the graph approaches, as the degree increases, some r-θ relationship quite different from the stereographic projection.

In order to see what this means, we did a rectification experiment. Figure 8(a) shows a fisheye lens image viewing a square grid pattern in approximately 30 degree direction, and Fig. 8(b) is the rectified perspective image, using the parameters of correction degree 5 in Table 1. The image is converted to a view as if observed by rotating the camera by 60 degrees to face the pattern (see Appendix for this computation). The black area near the left boundary corresponds to 95 degrees or more from the optical axis. Thus, we can obtain a correct perspective image to the very boundary of the view.

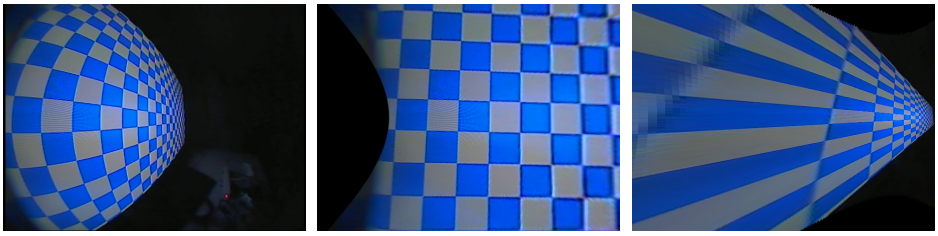
Table 1 Computed parameters for each correction degree.

degree	0	1	2	3	4	5
u_0	−1.56761	−1.56968	−1.57134	−1.59349	−1.59385	−1.60081
v_0	0.517809	0.488646	0.430145	0.423562	0.422805	0.426097
f	146.647	149.572	148.112	146.727	146.796	146.501
$a_1/10^{-2}$	—	0.646891	−0.305581	−1.41589	−1.68775	−1.80506
$a_2/10^{-3}$	—	—	2.39013	7.57212	11.1001	9.34256
$a_3/10^{-4}$	—	—	—	8.05471	22.2419	0.387907
$a_4/10^{-5}$	—	—	—	—	18.0654	61.6627
$a_5/10^{-6}$	—	—	—	—	—	0.935994



(a) (b) (c)

Figure 7 The dependence of the distance r (pixels) from the focal point on the incidence angle θ (degree) obtained by (a) using the collinearity, parallelism, and orthogonality constraints; (b) using only the collinearity constraints; (c) using the collinearity and parallelism constraints.



(a) (b) (c)

Figure 8 (a) Fisheye lens image viewing a square grid pattern in approximately 30 degree direction. (b) Rectified perspective image to be observed if the camera is rotated by 60 degrees to face the pattern. (c) Similarly rectified image using a spurious solution.

For comparison, Fig. 8(c) shows the result obtained by the same procedure using the spurious solution. We can see that collinear points are certainly rectified to be collinear and parallel lines to be (a skewed view of) parallel lines. The existence of such a spurious solution has not been known in the past collinearity-based work [1, 7, 11, 12, 14, 16]. Spurious solutions may be prevented by using auxiliary information such as vanishing point estimation [2, 4, 10, 11] or antipodal point extraction [13]. As we see, however, the spurious solution does not arise if the orthogonality constraint is imposed in addition.

The top-left of Fig. 9 is an image of a street scene taken from a moving vehicle with a fisheye lens camera mounted below the bumper at the car front. The top-right of Fig. 9 is the rectified perspective images. The second and third rows of Fig. 9 show the rectified perspective images to be observed if the camera is rotated by 90° left, right, up and down, confirming that we are really seeing more than 180° angles of view. Using a fisheye lens camera like this, we can detect vehicles approaching from left and/or right or create an image as if viewing the road from above the car.

9. Concluding remarks

We have presented a new technique for calibrating ultra-wide fisheye lens cameras. The basic principle of our calibration is the imposition of the constraint that collinear points be rectified to collinear, parallel lines to parallel, and orthogonal lines to be orthogonal. Exploiting the fact that line fitting reduces to eigenvalue problems, we optimized the constraint by invoking the perturbation theorem, as suggested by Onodera and Kanatani [14] in 1992. Then, we derived a Levenberg-Marquardt procedure for it. By experiments, we have found that a *spurious solution* exists if the collinearity constraint alone is used or even combined with the parallelism constraint. However, we have shown that by incorporating the orthogonality constraint an accurate calibration is done without using any auxiliary information such as vanishing point estimation [2, 4, 10, 11]. Finally, we have shown a real image example using a vehicle-mounted fisheye lens camera.

Acknowledgments. The authors thank Ryota Moriyasu of ERD Corporation for helping the experiments. This

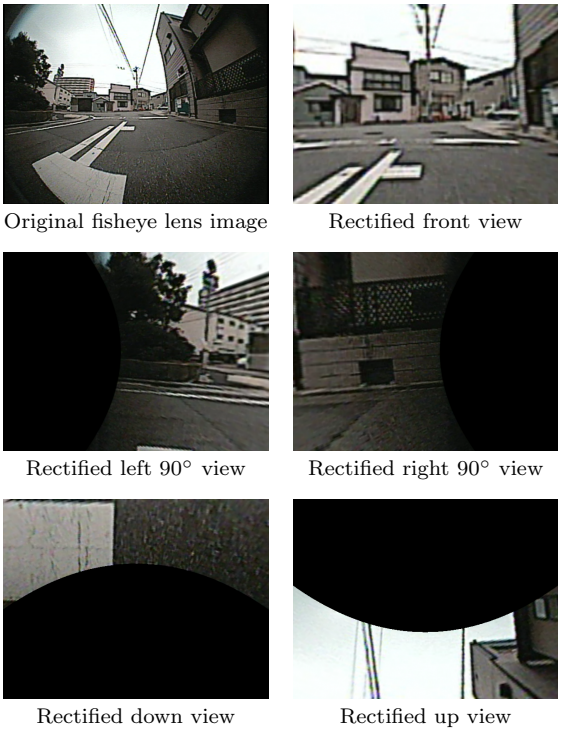


Figure 9 Fisheye lens image of an outdoor scene taken from a moving vehicle, rectified front images, and rectified images after virtually rotating the camera to left, right, up, and down.

work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology, Japan, under a Grant in Aid for Scientific Research (C 21500172).

References

[1] F. Devernay and O. Faugeras, Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments, *Machine Vision Appl.*, **13**-1 (2001-8), 14–24.

[2] R. Hartley and S. B. Kang, Parameter-free radial distortion correction with center of distortion estimation, *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**-8 (2007-8), 1309–1321.

[3] J. Heikkilä, Geometric camera calibration using circular control points, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-10 (2000-10).

[4] C. Hughes, P. Denny, M. Glavin and E. Jones, Equidistant fish-eye calibration and rectification by vanishing point extraction, *IEEE Trans. Patt. Anal. Mach. Intell.*, **32**-12 (2010-12), 2289–2296.

[5] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier, Amsterdam, the Netherlands, 1996; reprinted Dover, New York, NY, U.S.A., 2005.

[6] J. Kannala and S. S. Brandt, A general camera model and calibration method for conventional, wide angle, and fisheye-lenses, *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**-8 (2006-8), 1335–1340.

[7] S. Kase, H. Mitsumoto, Y. Aragaki, N. Shimomura and K. Umeda, A method to construct overhead view images using multiple fish-eye cameras (in Japanese), , *J. JSPE*, **75**-2 (2009-2), 251–255.

[8] H. Komagata, I. Ishii, A. Takahashi, D. Wakabayashi and H. Imai, A geometric calibration method of internal camera parameters for fish-eye lenses (in Japanese), *IEICE Trans. Inf. & Syst.*, **J89-D**-1 (2006-1), 64–73.

[9] Y.-C. Liu, K.-Y. Lin, and Y.-S. Chen, Bird's eye view vision system for vehicle surrounding monitoring, *Proc. 2nd Int. Workshop, Robvis2008*, February 2008, Auckland, New Zealand, pp. 207–218.

[10] M. Nakano, S. Li and N. Chiba, Calibration of fish-eye camera for acquisition of spherical image (in Japanese), *IEICE Trans. Inf. & Syst.*, **J89-D-II**-9 (2005-9), 1847–1856.

[11] M. Nakano, S. Li and N. Chiba, Calibrating fisheye camera by stripe pattern based upon spherical model (in Japanese), , *IEICE Trans. Inf. & Syst.*, **J89-D**-1 (2007-1), 73–82.

[12] R. Okutsu, K. Terabayashi, Y. Aragaki, N. Shimomura, and K. Umeda, Generation of overhead view images by estimating intrinsic and extrinsic camera parameters of multiple fish-eye cameras, *Proc. IAPR Conf. Machine Vision Applications*, May 2009, Yokohama, Japan, pp. 447–450.

[13] R. Okutsu, K. Terabayashi and K. Umeda, Calibration of intrinsic parameters of a fish-eye camera using a sphere (in Japanese), *IEICE Trans. Inf. & Syst.*, **J89-D**-12 (2010-12), 2645–2653.

[14] Y. Onodera and K. Kanatani, Geometric correction of images without camera registration (in Japanese), *IEICE Trans. Inf. & Syst.*, **J75-D-II**-5 (1992-5), 1009–1013.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge, U.K., 1992.

[16] F. Swaminathan and S. K. Nayar, Nonmetric calibration of wide-angle lenses and polycameras, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-10 (2000-10), 1172–1178.

[17] Z. Zhang, Flexible new technique for camera calibration, *IEEE Trans. Patt. Anal. Mach. Intell.*, **22**-11 (2000-11), 1330–1334.

Appendix: Image rectification

The rectification to a perspective image is done by the following procedure:

1. For each pixel $(\bar{x}, \bar{y})$, compute the incident angle θ by

$$\theta = \tan^{-1} \frac{\sqrt{X^2 + Y^2}}{Z} = \tan^{-1} \frac{\sqrt{R^2 - Z^2}}{Z}, \quad (49)$$

and compute the corresponding 3-D point (X, Y, Z) by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{R}{\sqrt{\bar{x}^2 + \bar{y}^2 + \bar{f}^2}} \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{f} \end{pmatrix}, \quad (50)$$

where the radius R is arbitrary (we may let $R = 1$).

2. Compute the corresponding pixel position (x, y) by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \frac{r(\theta)}{\sqrt{R^2 - Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix}, \quad (51)$$

and copy its pixel value to $(\bar{x}, \bar{y})$. If (x, y) are not integers, its value is interpolated from surrounding pixels.

To generate a perspective image to be observed by rotating the camera by $\boldsymbol{R}$ (rotation matrix) is obtained if Eq. (50) is replaced by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{R}{\sqrt{\bar{x}^2 + \bar{y}^2 + \bar{f}^2}} \boldsymbol{R} \begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{f} \end{pmatrix}. \quad (52)$$