Deep Single Fisheye Image Camera Calibration for Over 180-degree Projection of Field of View

Nobuhiko Wakai
Panasonic Corporation
wakai.nobuhiko@jp.panasonic.com

Takayoshi Yamashita
Chubu University
takayoshi@isc.chubu.ac.jp

Abstract

We propose a learning-based calibration method for trigonometric function models that represent distortion with over $180°$ *projection of field of views. Unlike perspective projection for less than* $180°$ *projection of field of views, fisheye projection such as equisolid angle projection is valid for whole world coordinates. To calibrate fisheye camera models, we define a new loss function based on camera projection effectively to optimize fisheye camera extrinsic (tilt and roll angles) and intrinsic (focal length) parameters. Our loss achieves small prediction errors throughout the ranges of parameters. Our results show that our method predicts precise fisheye camera parameters compared with conventional polynomial function models for radial distortion. This work is the first to calibrate a fisheye camera model including extrinsic and intrinsic parameters for over* $180°$ *projection of field of views from a single image to our knowledge.*

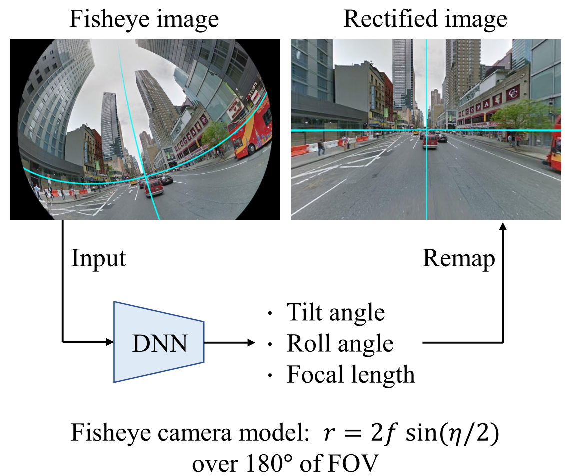

Figure 1: Concept illustration of our work. Our network predicts fisheye camera parameters using the trigonometric function model for over $180°$ of field of views to obtain rectified images by remapping. Cyan lines indicate the vertical and horizontal lines in each of the images.

1. Introduction

Fisheye cameras are widely used as a surveillance camera, a sensor for vehicles, advanced driver assistance systems, and robots. We focus on sensing applications such as surveillance and object detection with below steps; 1) users put cameras anywhere, 2) calibration without specific objects from a single image, 3) visualization or recognition using rectified images. Note that capturing images without calibration objects is substantially a single image of the background in calibration. The procedure is one of the typical settings independent on the application details. Although the application cameras need calibrating by users, these steps enable us to employ the sensing applications for wide usage indoors and outdoors. Further, these steps are used for cameras fixed to buildings or poles, *i.e.*, these cameras are hard to detach.

To remove the distortion or to measure distance using stereo cameras, it is essential to calibrate cameras. In fisheye cameras, captured images have large distortion especially near the edge of images. This distortion leads to degrading calibration accuracy. Geometric-based calibration methods require calibration objects such as chess boards and sequential images for motion-based methods. Further, we need to carry out a lot of calibration steps. This geometric-based calibration method requires strong constraint extracted using geometric information such as vanishing points and lines based on the calibration object. Although we must control calibration environment, the geometric-based methods have been well-established. In contrast, learning-based calibration methods achieve to calibrate several camera models from a single image. This learning-based method has an advantage of robustness on image illumination condition and scene. However, it is difficult to calibrate fisheye cameras regarding both extrinsic and intrinsic parameters due to large distortion including over $180°$ projection of field of views (FOV). In addition, high dimension polynomial function is unstable in optimization of learning-based methods. Despite the advantage of large FOV, the learning-based calibration method using a single image for fisheye cameras has been less discussed in the literature.

In this work, we propose a fisheye camera calibration method to predict extrinsic parameters (tilt and roll angles) and focal length in a trigonometric function model on the basis of a feature extractor composed of convolutional neural networks and regressors for individual parameters from a single image. We consider extrinsic parameters based on horizontal lines and do not recover full rotation matrix and translation vector, so-called *place recognition*.

The major contributions of this paper are two-fold. First, we propose a learning-based calibration method for fisheye cameras considering over 180° projection of FOV instead of perspective projection. Second, we demonstrate that a new loss function based on camera projection effectively to optimize fisheye camera extrinsic and intrinsic parameters.

This work is the first to calibrate fisheye camera models including extrinsic and intrinsic parameters considering over 180° projection of FOV from a single image to the best of our knowledge.

2. Related works

Camera calibration has been of interest because calibrated cameras are commonly adopted in various applications including surveillance and robotics. It has been well understood that calibration methods use given correspondences in world coordinates and image coordinates from calibration objects of a cubic [26] or planes [32]. Moreover, learning-based methods have been developed using single or multiple images in the wild, and these methods are based on convolutional neural networks. Camera parameters are composed of two elements: extrinsic parameters (rotation and translation) and intrinsic parameters (sensor and distortion parameters). In this paper, we focus on the learning-based calibration methods.

Learning-based methods for only extrinsic parameters were proposed for narrow-view cameras, *i.e.*, non-fisheye cameras, [8, 18, 24, 28, 29] and panoramic 360° images [2]. Image distortion is not negligible in fisheye cameras, and these methods using narrow-view cameras are not applied for fisheye cameras. In addition, Davidson's method [2] does not deal with non-panoramic fisheye images.

In addition to extrinsic parameters, calibration methods including focal length for narrow-view cameras were proposed using depth estimation [1, 4] or room layout [21]. It is useful for cameras based on perspective projection less than 180° projection of FOV because these methods calibrate parameters with focal length. However, these methods are not effective for fisheye cameras with over 180° projection of FOV.

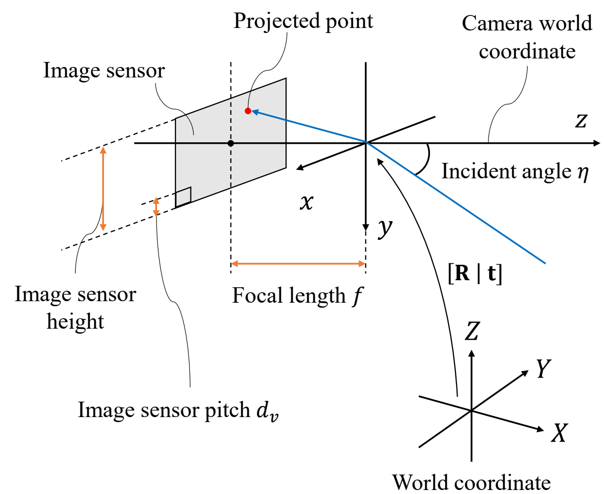

Figure 2: Illustration of camera parameters for projection from a blue incident ray.

Precisely to estimate distortion, calibration methods excluding extrinsic parameters were proposed using segmentation information [31], straight lines [30], or ordinal distortion of part of images [13]. Although these calibration methods have achieved precisely to calibrate intrinsic parameters including principal points, these methods cannot predict extrinsic parameters and are suit for only image undistortion.

A pioneer calibration method for extrinsic and intrinsic parameters including distortion was proposed by López-Antequera *et al*. [15]. This method used a polynomial function model based on perspective projection with a trainable distortion parameter k_1 and a distortion parameter k_2 calculated using a quadratic function depending on the parameter k_1. The method can address only less than 180° projection of FOV because the camera model is based on the perspective projection. Therefore, it is not adapted for fisheye cameras considering over 180° of FOV.

As mentioned above, conventional learning-based calibration methods do not consider over 180° of FOV because these methods are based on the perspective projection.

3. Proposed method

This section begins with describing camera projection models for clarifying our setting and notations of mathematical symbols. We then depict our deep neural network architecture. Finally, we explain our loss for the learning approach.

3.1. Camera models

Camera models express the mapping from the world coordinates to image coordinates in Fig. 2. The projection first converts the world coordinates to the camera coordinates by a 3×3 rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, as a whole called extrinsics $[\,\mathbf{R} \mid \mathbf{t}\,]$.

For a nonlinear projection model, the mapping can be written in a general form using a nonlinear function $\Gamma:\ \mathbb{R}^4 \rightarrow \mathbb{R}^3$ as

$$\tilde{\mathbf{u}} = \Gamma\left([\,\mathbf{R} \mid \mathbf{t}\,]\,\tilde{\mathbf{p}}\right), \tag{1}$$

in which $\mathbf{u} \in \mathbb{R}^2$ and $\mathbf{p} \in \mathbb{R}^3$ represents a point in the image and world coordinates, respectively, and a tilde over the vectors denotes the corresponding homogeneous coordinates.

For radial distortion [19] and fisheye lens distortion [25], the projector Γ can be expressed by a matrix that contains a nonlinear function, whose argument is $[\,\mathbf{R} \mid \mathbf{t}\,]\,\tilde{\mathbf{p}}$, as

$$\tilde{\mathbf{u}} = \begin{bmatrix} \gamma f/d_u & 0 & c_u \\ 0 & \gamma f/d_v & c_v \\ 0 & 0 & 1 \end{bmatrix} [\,\mathbf{R} \mid \mathbf{t}\,]\,\tilde{\mathbf{p}}, \tag{2}$$

with focal length f [mm], image sensor pitch (length per pixel) (d_u, d_v) [mm/pixel], and the principal point (c_u, c_v). The subscripts of u and v represent the horizontal and vertical direction of image coordinates, respectively.

Tsai's polynomial function [26] is an example of a polynomial function for the distortion represented as

$$\gamma = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \cdots, \tag{3}$$

where r denotes the distance from the principal point, and $\kappa_1, \kappa_2, \ldots$ are the polynomial coefficients. Note that the polynomial function is applied after perspective projection. Therefore, only less than 180° projection of FOV is valid.

Trigonometric function models γ for fisheye lenses [11, 20] are:

$$\gamma = \begin{cases} 2\tan(\eta/2) & \text{stereographic projection} \\ \eta & \text{equidistance projection} \\ 2\sin(\eta/2) & \text{equisolid angle projection} \\ \sin\eta & \text{orthogonal projection} \end{cases}, \tag{4}$$

where the argument $\eta = \arctan\left(z^{-1}\sqrt{x^2+y^2}\right)$ and $[x, y, z, 1]^\top = [\,\mathbf{R} \mid \mathbf{t}\,]\,\tilde{\mathbf{p}}$. Over 90°-incident angle η, *i.e.*, over 180° of FOV, is valid except for the orthogonal projection in Eq. (4).

Although there is a generalized camera model [3] including fisheye camera models instead of trigonometric function in Eq. (4), this generalized camera model requires several parameters to represent distortion. Therefore, we use the trigonometric function models in Eq. (4) for fisheye camera calibration efficiently to train deep neural networks.

3.2. Network architecture

We use DenseNet-161 [7] pretrained by ImageNet [22] for the image feature extractor of our network and details as follows.

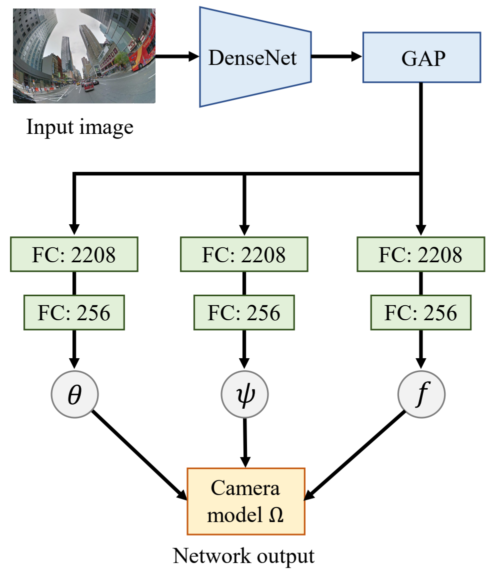

Figure 3: Our network architecture composed of a DenseNet feature extractor and regressors to predict camera parameters.

First, DenseNet extracts image features with 2208 channels from a single image with 224×224 pixels. These image features are employed using global average pooling (GAP) [14] to obtain a feature vector of 2208 dimension. Second, Normalized parameters of θ, ψ, and f are predicted using three individual regressors composed of a 2208-channel fully-connected (FC) layer with ReLU activation [12] and a 256-channel FC layer with sigmoid activation to predict the individual parameters in Fig. 3. In addition, batch normalization [10] is applied for each FC layer initialized using He's method [5]. Finally, predicted denormalized parameters are used for the predicted camera model Ω.

Previous works showed that scaling to 224×224 pixels for input images is appropriate transformation even though original images are not square [6, 15]. As follows this transformation, we scale input images.

3.3. Non-grid bearing loss

We define the non-grid bearing loss for training our network to calibrate fisheye cameras in Fig. 4. The bearing loss was proposed by López-Antequera *et al.* [15], and the loss was defined using the distance in the unit sphere of the world coordinates from standard-grid image coordinates projected by camera parameters. These standard-grid points outer the image circle are invalid for projection in fisheye cameras. Further, the grid points are not balanced for fisheye images because fisheye images have large distortion in image coordinates. Therefore, we define the non-grid bearing loss without standard-grid image coordinates described below.

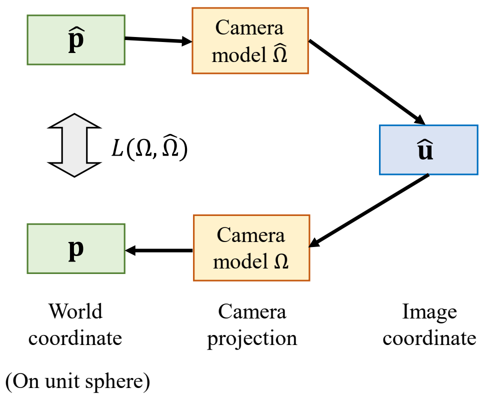

Figure 4: Non-grid bearing loss definition based on the camera projection using predicted and ground-truth parameters.

First, uniform world coordinates $\hat{\mathbf{p}}$ of n points ($n = 32,400$ for experiments) on the unit hemisphere within 90°-incident angles are projected to the image coordinates $\hat{\mathbf{u}}$ using ground-truth (GT) camera parameters. Second, these n points of image coordinates are projected to the unit sphere as the world coordinates $\mathbf{p}$ using predicted camera parameters. Finally, we evaluate Euclidean distance between $\mathbf{p}$ and $\hat{\mathbf{p}}$.

We show the equation of the non-grid bearing loss L as,

$$L(\Omega, \hat{\Omega}) = \frac{1}{n} \sum_{i=1}^{n} \text{Huber}(||\mathbf{p_i} - \hat{\mathbf{p}}_\mathbf{i}||_2), \tag{5}$$

where Ω and $\hat{\Omega}$ are predicted and ground-truth camera parameters, respectively. Additionally, $\mathbf{p}$ and $\hat{\mathbf{p}}$ are the world coordinates projected by Ω and $\hat{\Omega}$, respectively. The Huber $(\bullet)$ denotes Huber loss function with $\delta = 1$, so-called smooth L_1 loss [9].

We define the total network loss L_{total} for training our network shown in,

$$L_{total} = w_\theta L_\theta + w_\psi L_\psi + w_f L_f,$$
$$\text{where} \begin{cases} L_\theta &= L(\Omega(\theta, \hat{\psi}, \hat{f}), \hat{\Omega}) \\ L_\psi &= L(\Omega(\hat{\theta}, \psi, \hat{f}), \hat{\Omega}) \\ L_f &= L(\Omega(\hat{\theta}, \hat{\psi}, f), \hat{\Omega}) \end{cases}, \tag{6}$$

w_θ, w_ψ, and w_f are joint weights of θ, ψ and f, respectively, and $\{\hat{\bullet}\}$ indicates the ground truth values. Since the camera projection sensitively depends on camera parameters, we use the joint loss L_{total} consisting a predicted parameter and ground truth parameters for the rest.

4. Experiments

For evaluating our method calibrating fisheye cameras, we conduct training and evaluating our network compared with conventional calibration methods. First, we describe experiment settings of dataset and parameters, and then we show experimental results.

Parameters	Distribution	Range or values
Pan ϕ	Uniform	$[0, 2\pi)$
Tilt θ	Mix	7/9 Normal, 2/9 Uniform
	Normal	$\mu = 0, \sigma = \pi/6$
	Uniform	$[-\pi/2, \pi/2]$
Roll ψ	Mix	7/9 Normal, 2/9 Uniform
	Normal	$\mu = 0, \sigma = \pi/6$
	Uniform	$[-\pi/2, \pi/2]$
Aspect ratio	Varying	{1/1 9%, 5/4 1%, 4/3 66%, 3/2 20%, 16/9 4%}
Focal length f	Uniform	[8.5, 15]

Table 1: Distribution of the camera parameters to make our train and validation sets. Units: f [mm]; ϕ, θ, and ψ [rad].

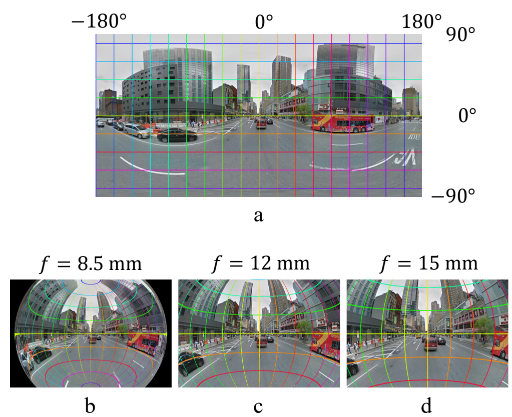

Figure 5: Example of rendered images. (a) An input panorama image [17] with grid lines every 20°. Rendered fisheye images in (b), (c), and (d) using focal length set to 8.5 mm, 12 mm, and 15 mm, respectively, and the aspect ratio of these images is 3/2.

4.1. Dataset

We use a large-scale dataset of outdoor panoramas named StreetLearn dataset (Manhattan 2019 subset) [17] artificially to make images using arbitrary camera parameters in Fig. 5. We render train, validation, and test images using these panorama images whose size is 1664×832 pixels as described below. First, StreetLearn dataset is divided into train (including validation) and test sets of $55,599$ and 161 images, respectively. We render 9 and 100 image patches for train and test sets, respectively. Train, validation, and test sets have $488,883$, $11,508$, and $16,100$ images, respectively because we use validation rate 0.023 for train and validation division. Second, we generate parameters with random distribution shown in Tab. 1. The dataset division and aspect ratio distribution are based on the previous work [15]. Since the zero-centered normal distribution rarely generates large values, we mix the normal distribution and uniform distribution to obtain large rotation for tilt and roll angles. Only uniform distribution is used for non-trainable pan angles.

In test set, we use uniform distribution to evaluate parameters considering large rotation and varying aspect ratios described below. We use uniform distribution $[-\pi/2, \pi/2]$ for tilt angle θ and roll angle ψ. In addition, the aspect ratios have varying {1/1 20%, 5/4 20%, 4/3 20%, 3/2 20%, 16/9 20%}. The test distribution of pan angle ϕ and f is the same to train distribution.

The dataset division and aspect ratio distribution are based on the previous work [15]. Additionally, we use the wide-range distribution of focal length for diagonal fisheye cameras and circumferential fisheye cameras capturing images with non-projection areas, *i.e.*, outer image circles in Fig. 5. In the circumferential fisheye images, we fill the outer image circle pixels with the mean value of the overall dataset.

4.2. Parameter and network settings

We use the camera model in Eq. (1) with the equisolid angle projection in Eq. (4). Since the translation vector $\mathbf{t}$ is arbitrary, we fix $\mathbf{t}$ as zero-vector. Additionally, we fix the principal points (c_u, c_v) set to the center of image to simplify the camera model. We assume that the image sensor height is $24\,\mathrm{mm}$ mimicking the full size image sensor because the scale factor depends on not only focal length but image sensor size. Note that this image sensor size is arbitrary for rectification, and the focal length is scaled by the sensor size. Image sensor pitch d_v in Eq. (1) is calculated using the image sensor height [mm] and image height [pixel]. Further, d_u is set to d_v on assumption of square pixels.

The pan angles are given for training and evaluation because the origin of pan angles is arbitrary in panorama images. Therefore, we focus on three trainable parameters of tilt angle θ, roll angle ψ, and focal length f in our method.

Our network is optimized by Adam optimizer using decoupled weight decay regularization [16] whose weight decay is 0.01. The initial learning rate is set to 1×10^{-5}, and this learning rate is multiplied by 0.1 at 11 epoch in Fig. 6. We use early stopping appropriately to finish training at 13 epoch. In addition, the batch size is 32, and all joint weights of w_θ, w_ψ, and w_f are set to $1/3$ for the non-grid bearing loss in Eq. (6).

4.3. Experimental results

There is no common evaluation way for single image camera calibration due to lack of consensus. Although there are several metrics for evaluation of camera models, it is difficult to evaluate fairly because the precision of camera calibration depends on application, *i.e.*, an image undistortion task requires small errors in image coordinates but a stereo measurement task requires small incident angle errors for stereo measurement. We follows previous works [6, 15] reporting error distribution. In addition, there is no conventional methods appropriately to compare our method because only our network can predict extrinsic parameters of camera rotation and intrinsic parameters in fisheye cameras for over 180° of FOV from a single image. Therefore, we first describe error distribution compared with ground truth parameters to show calibration accuracy. Second, we partially compare our method and conventional methods due to the difference of network output, *i.e.*, some conventional methods predict only intrinsic parameters.

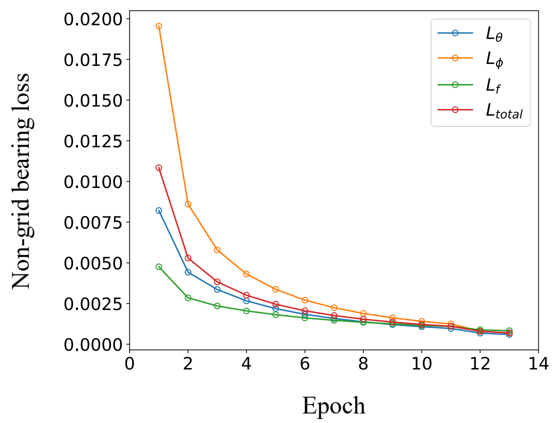

Figure 6: Non-grid bearing train loss. All joint weights of w_θ, w_ψ, and w_f are set to $1/3$.

4.3.1 Error distribution of our network

We show the error distribution of our network using the test set. The distribution is mainly plotted on the diagonal lines in tilt angle θ, roll angle ψ, and focal length f in Fig. 7. This trend means that predicted parameters are corresponded to ground truth parameters throughout x-axis of parameter ranges. Additionally, we show the absolute errors between ground-truth and predicted values among these parameters in Fig. 8. This distribution of these absolute errors represents that there are small errors throughout parameter ranges. Unlike previous work [15], tilt angle θ and roll angle ψ errors are not increased in large rotation angles because of our loss and our mixed distribution using not only normal distribution but uniform distribution.

Figure 6 shows each non-grid bearing loss in tilt angle θ, roll angle ψ, focal length f, and L_{total}. At the beginning of training, the non-grid bearing loss of roll angle L_ϕ has the largest loss compared with tilt angle L_θ and focal length L_f. However, training using our joint loss L_{total} optimizes parameters, and each parameter has the same mag-

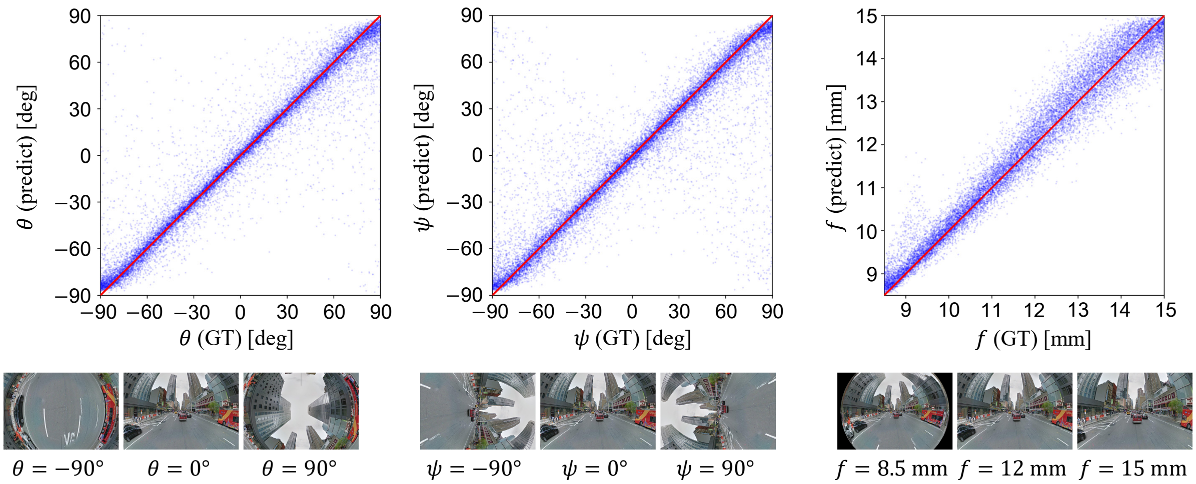

Figure 7: Error distribution on the test set of $16,100$ images. The horizontal axis indicates ground truth values of parameters. The vertical axis indicates predicted parameters. The diagonal red lines indicate the perfect prediction. The bottom images are examples of rendered images using notated camera parameters.

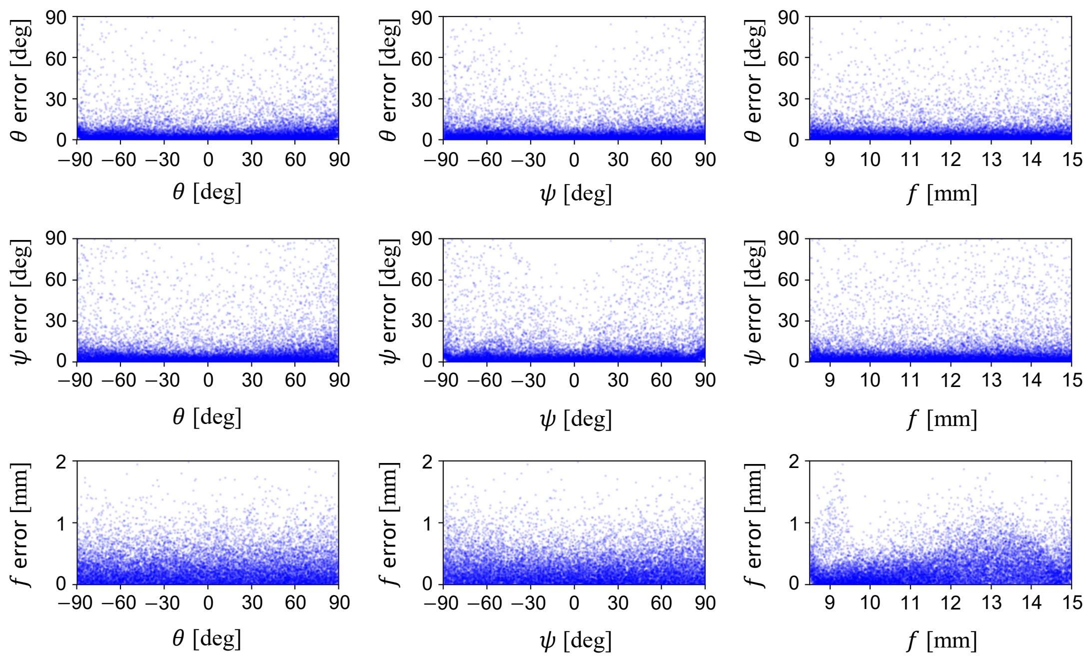

Figure 8: Errors on the test set of $16,100$ images. The horizontal axis denotes ground truth values while the vertical axis denotes the absolute error between ground-truth and predicted values.

nitude of loss after convergence. Therefore, our joint loss L_{total} works effectively to optimize fisheye camera parameters and leads to achieving precise calibration for extrinsic and intrinsic parameters.

For predicting tilt angle θ, roll angle ψ, and polynomial distortion coefficient in perspective projection, López-Antequera *et al.* [13] proposed a learning-based method. Although the intrinsic parameters of López-Antequera's method are different from our method, extrinsic parameter representation is equivalent. Therefore, we compare the extrinsic parameters in López-Antequera's method and our methods as shown below. First, we train the network of López-Antequera's method using StreetLearn dataset [17] divided into train, validation, and test set followed in Sec. 4.1. Although we train López-Antequera's network using the distribution of train set provided in the corresponding paper, the distribution of test set is the same distribution in Sec. 4.1 for evaluation.

In our method, the absolute errors between ground-truth and predicted parameters in tilt angle θ, roll angle ψ, and focal length f are 6.62 ± 13.21 [deg], 9.34 ± 19.89 [deg], and 0.276 ± 0.257 [mm] (Mean $\pm$ S.D.), respectively. In López-Antequera's method, the absolute errors in tilt angle θ and roll angle ϕ are 31.78 ± 30.03 [deg] and 45.25 ± 25.91 [deg], respectively. Although we train the network of López-Antequera's method in the corresponding paper except for using StreetLearn dataset [17] consisting of various street scene, it seems that it is difficult for López-Antequera's method to train the networks using StreetLearn dataset even if the training is converged. Therefore, our method achieves small errors in rotation parameters compared with López-Antequera's method throughout angle ranges. Note that intrinsic parameter evaluation is described later in Sec. 4.3.3.

4.3.2 Reprojection error

It is well-known to evaluate camera parameters using reprojection errors in geometric-based calibration methods. The reprojection errors represent the calibration accuracy of both extrinsic and intrinsic parameters. In learning-based calibration method using a single image, there is no explicit ground-truth points in the world coordinate. For learning-based methods, we evaluate extrinsic and intrinsic parameters using reprojection errors described below. First, uniform world coordinates $\hat{\mathbf{p}}$ of n $(= 32,400)$ points on the unit hemisphere within $90°$-incident angles are projected to the image coordinates $\hat{\mathbf{u}}$ using the ground-truth camera parameter $\hat{\Omega}$. Similarly, the world coordinates $\hat{\mathbf{p}}$ are projected to the image coordinates $\mathbf{u}$ using predicted camera parameter Ω. We show the reprojection error ϵ as,

$$\epsilon(\Omega, \hat{\Omega}) = \frac{1}{n} \sum_{i=1}^{n} ||\mathbf{u_i} - \hat{\mathbf{u}}_\mathbf{i}||_2^2. \tag{7}$$

In the test set, the reprojection errors of our method and López-Antequera's method described in Sec. 4.3.1 are 17.99 ± 15.44 and 42.14 ± 48.02 pixels, respectively. Note that we clamp distance between $\mathbf{u}$ and $\hat{\mathbf{u}}$ using half image height (112 pixels) because reprojection errors may cause quite large values. In addition to rotation errors, our method has small reprojection errors compared with López-Antequera's method that has large reprojection errors due to large errors of roll angle ψ.

4.3.3 Comparison using PSNR and SSIM

We use the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [27] for intrinsic parameter evaluation. In image rectification task, extrinsic parameters are ignored because the image rectification is employed using only intrinsic parameters such as focal length and distortion coefficients. In general, the image rectification is used for evaluation of non-fisheye cameras because large incident angle cannot be projected to rectified images due to over $180°$ of FOV in fisheye cameras.

In methods predicting only intrinsic parameters for rectification, Yin *et al*. [31] proposed a learning-based method regarding image context of segmentation, and Liao *et al*. [13] proposed a learning-based method using ordinal distortion in parts of images to use alternative representation compared with Yin's method. In addition, the state-of-the-art geometric-based calibration method was proposed by Santana-Cedrés *et al*. [23] for rectification using lines. These baseline models described above are realized according to the implementation details provided in corresponding papers.

For fisheye evaluation, we render images using the test set described below. First, we use a pinhole camera with $120°$ of FOV for remapping to obtain rectified images from the test set and ground truth parameters. In Yin's method and Liao's methods, we employ center cropping for the input images before feeding them to networks because these methods require square input images. Second, we calculate PSNR and SSIM using rectified images of the ground truth and prediction.

Table 2 shows comparison of PSNR and SSIM in our test set. Our method outperforms conventional methods in both PSNR and SSIM. Our method and López-Antequera's method have higher accuracy compared with methods predicting only intrinsic parameters. Therefore, training using extrinsics and intrinsics parameters simultaneously probably leads to improving accuracy. Note that we exclude Santana-Cedrés's method for quantitative evaluation because it does not work in many images because the line detector fails to extract lines.

The qualitative rectification results on our test dataset generated by our method and the others are shown in Fig. 9. Our method obtains overall the most similar to the ground truth images even if cameras are rotated with large angles.

As described above, our method precisely calibrates both extrinsic and intrinsic parameters for fisheye cameras with over $180°$ of FOV.

5. Conclusion

We have described our learning-based calibration method using the trigonometric function model for extrinsic and intrinsic parameters in fisheye cameras. Effectively to calibrate fisheye camera, we proposed the non-grid bearing loss to represent distance errors on unit sphere projected by camera parameters. The main result of this paper is that our method calibrates not only intrinsic parameters but extrinsic parameters from a single image for over $180°$ of FOV. In addition, our method precisely calibrates parameters compared with both conventional geometric-based and learning-based methods. Evaluation using various camera models is our future works.

Method	Learning	Extrinsics	Intrinsics	Projection	Over 180° FOV	PSNR ↑	SSIM ↑
Santana-Cedrés [23][1]			✓	Perspective		–	–
Yin [31]	✓		✓	Fisheye [2]	✓	15.29 ± 1.78	0.3344 ± 0.1213
Liao [13]	✓		✓	Perspective		15.52 ± 1.98	0.3859 ± 0.1173
López-Antequera [15]	✓	✓	✓	Perspective		16.92 ± 3.87	0.4555 ± 0.1769
Ours	✓	✓	✓	Fisheye	✓	$\mathbf{21.72 \pm 5.56}$	$\mathbf{0.6124 \pm 0.2078}$

[1] Exclusion for evaluation due to failure of line detection in many images.
[2] Using generalized fisheye camera models.

Table 2: Comparison of conventional methods and our method using the test set.

Input GT Santana-Cedrés Yin Liao López-Antequera Ours

Figure 9: Qualitative results on our test images. We show the input image, the ground truth image, and the results of compared methods: Santana-Cedrés [23], Yin [31], Liao [13], López-Antequera [15], and our method from left to right.

References

[1] Y. Chen, C. Schmid, and C. Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 7062–7071, 2019. 2

[2] B. Davidson, M. S. Alvi, and J. F. Henriques. 360^o Camera alignment via segmentation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2

[3] D. B. Gennery. Generalized camera calibration including fish-eye lenses. *International Journal of Computer Vision (IJCV)*, 68:239–266, 2006. 3

[4] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 8976–8985, 2019. 2

[5] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 3

[6] Y. Hold-Geoffroy, K. Sunkavalli, J. Eisenmann, M. Fisher, E. Gambaretto, S. Hadap, and J. Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2354–2363, 2018. 3, 5

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 3

[8] Z. Huang, Y. Xu, J. Shi, X. Zhou, H. Bao, and G. Zhang. Prior guided dropout for robust visual localization in dynamic environments. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2791–2800, 2019. 2

[9] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964. 4

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[11] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(8):1335–1340, 2006. 3

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[13] K. Liao, C. Lin, and Y. Zhao. A deep ordinal distortion estimation approach for distortion rectification. *arXiv preprint arXiv:2007.10689*, 2020. 2, 6, 7, 8

[14] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2014. 3

[15] M. López-Antequera, R. Marí, P. Gargallo, Y. Kuang, J. Gonzalez-Jimenez, and G. Haro. Deep single image camera calibration with radial distortion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11809–11817, 2019. 2, 3, 5, 8

[16] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2019. 5

[17] P. Mirowski, A. Banki-Horvath, K. Anderson, D. Teplyashin, K. M. Hermann, M. Malinowski, M. K. Grimes, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell. The StreetLearn environment and dataset. *arXiv preprint arXiv:1903.01292*, 2019. 4, 6, 7

[18] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang. Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 52–61, 2020. 2

[19] G. V. Puskorius and L. A. Feldkamp. Camera calibration methodology based on a linear perspective transformation error model. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1858–1860 vol.3, 1988. 3

[20] F. S. Ray. *Applied photographic optics*. Focal Press Oxford, 1994. 3

[21] L. Ren, Y. Song, J. Lu, and J. Zhou. Spatial geometric reasoning for room layout estimation via deep reinforcement learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[23] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, and L. Alvarez. An iterative optimization algorithm for lens distortion correction using two-parameter models. *Image Processing On Line*, 6:326–364, 2016. 7, 8

[24] M. R. U. Saputra, P. Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni. Distilling knowledge from a deep pose regressor network. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 263–272, 2019. 2

[25] S. Shah and J. K. Aggarwal. A simple calibration procedure for fish-eye (high distortion) lens camera. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3422–3427 vol.4, 1994. 3

[26] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987. 2, 3

[27] Z. Wang and A. C. Bovik. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. 7

[28] W. Xian, Z. Li, N. Snavely, M. Fisher, J. Eisenman, and E. Shechtman. UprightNet: Geometry-aware camera orientation estimation from single images. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 9973–9982, 2019. 2

[29] F. Xue, X. Wang, Z. Yan, Q. Wang, J. Wang, and H. Zha. Local supports global: Deep camera relocalization

with sequence enhancement. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2841–2850, 2019. 2

[30] Z. Xue, N. Xue, G. Xia, and W. Shen. Learning to calibrate straight lines for fisheye image rectification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1651, 2019. 2

[31] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao. FishEyeRecNet: A multi-context collaborative deep network for fisheye image rectification. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2, 7, 8

[32] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(11):1330–1334, 2000. 2