DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras

Oleksandr Bogdan*
CML, KAIST
Daejeon, Republic of Korea

Viktor Eckstein*†
Karlsruhe Institute of Technology
Karlsruhe, Germany

Francois Rameau
RCV Lab, KAIST
Daejeon, Republic of Korea

Jean-Charles Bazin
CML, KAIST
Daejeon, Republic of Korea

ABSTRACT

Calibration of wide field-of-view cameras is a fundamental step for numerous visual media production applications, such as 3D reconstruction, image undistortion, augmented reality and camera motion estimation. However, existing calibration methods require multiple images of a calibration pattern (typically a checkerboard), assume the presence of lines, require manual interaction and/or need an image sequence. In contrast, we present a novel fully automatic deep learning-based approach that overcomes all these limitations and works with a single image of general scenes. Our approach builds upon the recent developments in deep Convolutional Neural Networks (CNN): our network automatically estimates the intrinsic parameters of the camera (focal length and distortion parameter) from a single input image. In order to train the CNN, we leverage the great amount of omnidirectional images available on the Internet to automatically generate a large-scale dataset composed of millions of wide field-of-view images with ground truth intrinsic parameters. Experiments successfully demonstrated the quality of our results, both quantitatively and qualitatively.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Camera calibration, focal length, lens distortion, deep learning, self-calibration, fisheye lens.

ACM Reference format:
Oleksandr Bogdan, Viktor Eckstein, Francois Rameau, and Jean-Charles Bazin. 2018. DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras. In *Proceedings of CVMP '18: European Conference on Visual Media Production, London, United Kingdom, December 13–14, 2018 (CVMP '18),* 10 pages.
https://doi.org/10.1145/3278471.3278479

*denotes joint first authorship with equal contribution.
†This work was completed during Viktor's internship at CML, KAIST.

CVMP '18, December 13–14, 2018, London, United Kingdom

ACM ISBN 978-1-4503-6058-6/18/12…$15.00
https://doi.org/10.1145/3278471.3278479

1 INTRODUCTION

Wide field-of-view (FOV) cameras permit to acquire images with wide angles and are typically equipped with a fisheye lens, such as the popular GoPro cameras. Thanks to their wide field-of-view, they are useful for several tasks related to visual media production, such as camera motion estimation and 3D reconstruction [Bazin et al. 2010, 2012; Häne et al. 2014; Lee et al. 2013; Liu et al. 2017; Schöps et al. 2017], human actions and 3D skeleton [Rhodin et al. 2016], as well as AR [Streckel et al. 2005]. To enable such applications, it is required to calibrate the cameras. Camera calibration refers to the estimation of their intrinsic parameters [Hartley and Zisserman 2004], and the two most important calibration parameters for wide FOV cameras are the focal length and distortion parameter. Due to their inherent distortion and specific projection models, wide FOV cameras require dedicated calibration methods, see for example [Antunes et al. 2017; Fitzgibbon 2001; Mei and Rives 2007; Melo et al. 2013; Micusík and Pajdla 2003; Scaramuzza et al. 2006; Swaminathan and Nayar 2000; Ying and Hu 2004; Ying and Zha 2008], among many others. However, existing calibration methods for wide FOV cameras have important limitations. For example, they require multiple observations of a calibration object (e.g., checkerboard [Gasparini et al. 2009; Mei and Rives 2007; Scaramuzza et al. 2006], dot pattern [Shah and Aggarwal 1994] or sphere [Ying and Zha 2008]), and/or require the observation of specific structures in the scene (e.g., lines or vanishing points in structured scenes [Antunes et al. 2017; Barreto and Araújo 2005; Bräuer-Burchardt and Voss 2001; Melo et al. 2013; Swaminathan and Nayar 2000; Ying and Hu 2004]), and/or require estimating the camera motion from multiple images [Fitzgibbon 2001; Kang 2000; Micusík and Pajdla 2003; Xiong and Turkowski 1997; Zhang 1996]. In practice, the most popular approach among the ones listed above is based on checkerboards [Gasparini et al. 2009; Mei and Rives 2007; Scaramuzza et al. 2006], which requires taking several images of a checkerboard. In summary, existing camera calibration methods are time-consuming (e.g., several images and manual process), cumbersome (e.g., use of a checkerboard), require strong assumptions on the scene (e.g., lines and vanishing points) and/or cannot work on single images.

In contrast, we propose a deep learning-based approach for wide FOV camera calibration that overcomes all these limitations: it

does not require any motion estimation, calibration target, several images or specific structure in the scene. Moreover it works on single input image of general scenes and thus, can also be applied on single images "in the wild"[1] (e.g., downloaded from the Internet).

Given an input image acquired by any wide FOV camera (e.g., action camera like GoPro, DSLR camera with high-quality fisheye lens, smartphone with casual clip-on fisheye lens or catadioptric camera), our approach estimates the focal length of the camera and the distortion parameter in a **fully automatic** manner.

Our approach builds upon the recent developments in deep learning: we use a deep Convolutional Neural Network (CNN) with Inception-V3 architecture [Szegedy et al. 2016]. Our work focuses on two main aspects. First, a major challenge to train the network is the need for numerous training examples. In our context, we need plenty of images with different intrinsic parameters (focal length and distortion). For this, we leverage the large collection of omnidirectional images available on the Internet, which allows us to automatically generate numerous wide FOV images with different focal lengths and amounts of distortion. In turn, we can generate a large-scale dataset composed of millions of wide FOV images with ground truth intrinsic parameters that we use to train the CNN. The second aspect is the comparison of different CNN architectures (e.g., single network, dual networks and sequential networks).

The two main contributions of this paper are the following:

- We present a CNN-based approach for automatic calibration of wide FOV cameras. Contrary to existing methods, our approach does not need calibration targets (e.g., checkerboard), lines, multiple views or motion estimation. Given a single image of a general scene, it automatically estimates the focal length and distortion parameter.
- An automatic approach for generating a large-scale dataset composed of millions of wide FOV images with ground truth intrinsic parameters to train a CNN.

To the best of our knowledge, our work is the first deep learning approach for automatic calibration of wide FOV cameras from a single image. We provide code and additional materials on our project website *http://cml.kaist.ac.kr/projects/DeepCalib*.

2 RELATED WORK

Existing calibration methods: Camera calibration aims to estimate the intrinsic parameters of the camera [Hartley and Zisserman 2004]. Several methods for wide FOV camera calibration have been proposed, and can be divided into four main categories. The most popular and widely used category is based on a known calibration target (typically a checkerboard) placed in the scene and observed under different viewpoints in several images [Gasparini et al. 2009; Mei and Rives 2007; Scaramuzza et al. 2006; Shah and Aggarwal 1994]. Other targets have also been studied, such as dot pattern [Shah and Aggarwal 1994] or sphere [Ying and Zha 2008]. The methods belonging to this category are usually the most accurate, for example because the features in the images (e.g., checkerboard corners) can be precisely detected in the images and the calibration target model is known beforehand. An important limitation is that they require a specific calibration target and several images. Therefore, they are cumbersome, time-consuming, and also not applicable to single images "in the wild".

The second category is based on the presence of geometric structures in the scene, typically lines [Barreto and Araújo 2005; Workman et al. 2016; Zhang et al. 2015] and vanishing points [Antunes et al. 2017; Hughes et al. 2010]. For instance, the approach developed by Barreto and Araújo [2005] needs an image containing at least three lines that are manually given by the user, and Antunes et al. [2017] rely on orthogonal vanishing points. Therefore, these methods are limited to structured man-made scenes containing lines, and thus cannot deal with general environments, like landscapes or natural scenes, and when a large portion of the image is covered by an object, such as a face close-up.

The third category is camera self-calibration which jointly estimates the intrinsic parameters of the camera and its motion from a sequence of images. For example, Fitzgibbon [2001] solves a radial fundamental matrix and tri-focal tensor which allows to estimate together the distortion parameter with the epipolar geometry between two or three successive images. This seminal work led to several extensions with different configurations, number of images and minimal solutions [Jiang et al. 2014; Micusík and Pajdla 2003]. The two main limitations of self-calibration are the requirement of several images and the need to perform camera motion estimation, which is still a major challenge in itself (e.g., point correspondences, repetitive texture, lighting changes and motion ambiguity).

The last category is based on deep learning. One of the first attempts is the approach of Mendonça et al. [2002], which uses neural network to compute the camera parameters given 3D point locations from a calibration target and their respective 2D observations. Recent deep learning methods aiming to estimate camera parameters from a single image without calibration target have been attempted. However, all these techniques are designed to partly solve the calibration problem. For example, DeepFocal [Workman et al. 2015] predicts only the focal length (no distortion estimation). It is trained using few images (around 7,000). In contrast, we solve a more general and challenging problem (focal length and distortion), and propose an efficient method for generating millions of training images. Rong et al. [2016] estimate only the radial distortion (no focal length). Thus, their approach can only be applied for image visual undistortion. In contrast, our work can estimate both focal length and distortion parameter, and thus can be used for a wider range of applications, such as image undistortion and 3D reconstruction Section 4). Moreover, their dataset is generated from perspective images, which leads to incomplete wide FOV images with occlusion margins. In contrast, we can generate complete images (without any margins), even with a wide FOV. Hold-Geoffroy et al. [2018] estimate the focal length and camera orientation (no distortion parameter). They train a CNN on images generated from panoramas using standard pinhole model, and thus, are limited to perspective cameras. In contrast, our approach estimates both focal length and distortion. Therefore, it can handle a much wider range of cameras, such as fisheye and catadioptric cameras; and it also enables additional applications, such as SfM and image undistortion. Overall, estimating both the focal length and distortion requires dedicated methods and investigations, such as the selection of the distortion model, automatic training dataset generation, and network architectures (see Section 3).

[1] "in the wild" as mentioned in several works, such as [Bell et al. 2014; Chen et al. 2016; Lin et al. 2012].

Image undistortion: A common task when using wide FOV cameras is image undistortion (also called rectification in some contexts), i.e., "remove" the distortion of the input image, so that the lines in the world appear straight in the output rectified images. This is typically used for display and visualization purposes to have a more "natural" look. When the intrinsic parameters are known, image undistortion can be conducted, for example using popular image processing libraries and softwares, such as OpenCV, ImageMagick, PTLens and Hugin. This requires to know the intrinsic parameters, which needs to conduct calibration. However, as discussed above, existing calibration methods have several limitations and might not be applicable (e.g., single image available).

For photography professionals, a popular approach for image undistortion is the Lens Correction filter in Adobe Photoshop. This tool requires some information about the camera and lens, such as the camera model, focal length and lens model. Therefore, it is not applicable for unlisted camera or lens models, as well as for images "in the wild", such as images downloaded from the Internet for which camera or lens information might be unknown. In these cases, Photoshop allows the users to manually undistort the image via interactive sliders, but it is a time-consuming task that has to be repeated for each camera/lens.

Another commercial tool is the Fisheye-Hemi plug-in for Photoshop [Fisheye-Hemi 2015]. It is used for fisheye image visualization and aims to "produce an aesthetically pleasing image" from an input fisheye image (as quoted in their website). However, the underlying visualization algorithm is proprietary, no technical details are provided and the estimated camera parameters are not returned. Due to the lack of public information, it is not totally clear if it actually aims to estimate the camera parameters. This tool is made for image visualization, not for camera calibration, i.e., it does not return the camera intrinsic parameters. An approach similar to this tool is the visualization method of Carroll et al. [2009] which aims to create a visually pleasing version of wide FOV images. They optimize a spatially-varying mapping to preserve local shape and maintain scene lines straight. However their method is also applicable only for image visualization, i.e., it does not compute the intrinsic parameters. Moreover, it requires the user to mark lines manually (about 20 lines on average, as stated in their paper), which is a time-consuming task that must be repeated for each image.

In summary, contrary to the above methods dedicated to image undistortion, our approach performs explicit camera calibration and runs automatically, and thus enables several applications, such as not only image undistortion, but also SfM and inserting virtual objects in AR images.

3 PROPOSED APPROACH

This section presents the main aspects of our approach: selection of the camera distortion model (Section 3.1), automatic generation of a large-scale dataset with ground truth intrinsic parameters (Section 3.2), and description of our network architectures (Section 3.3).

3.1 Projection and distortion model

Wide FOV cameras require specific projection models to map a 3D world point to the image. Various models have been developed. The most common one is the Brown-Conrady's model [Brown 1971]. It

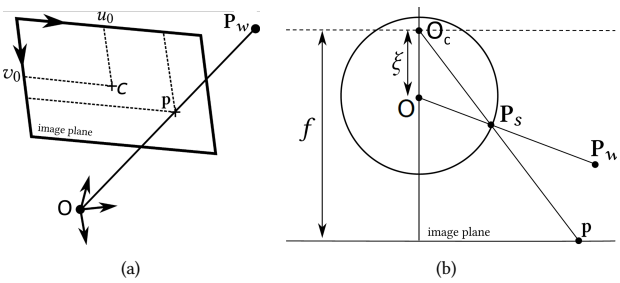


Figure 1: (a) pinhole camera model [Hartley and Zisserman 2004] and (b) unified spherical model [Barreto 2006; Mei and Rives 2007].

is particularly effective to approximate reasonably small distortions via a polynomial function. Despite its popularity, this model has several important limitations. For example, the types of cameras which can be modeled with this representation are limited. For instance, it is not suitable for wide FOV cameras due to their large inherent distortions [Sturm et al. 2011]. In addition, it is well known that this model is hardly reversible [Sturm et al. 2011].

Another popular model is the division model [Fitzgibbon 2001] to represent fisheye cameras. However, it has been designed only for fisheye lens and is not recommended for standard cameras. Similarly to Brown's model, the division model is theoretically impossible to revert. While it is possible to *approximate* the inversion, this may negatively affect the training process, particularly for large distortions [Tang et al. 2017] (see supplementary material).

In this paper, we opted for the unified spherical model [Barreto 2006; Mei and Rives 2007] for several reasons. First, it is fully reversible; second, it can handle a very large range of distortions (from none and small to very large); and third, both the projection and back-projection processes admit closed-form solution which can be computed very efficiently. This is particularly interesting for GPU image generation where this type of computation is faster than look-up-table [Häne et al. 2014]. Moreover, the spherical model is compatible with a wider range of cameras than other models, for example perspective, wide-angle, fisheye and catadioptric cameras. Lastly, it is particularly convenient for our application since it involves a single distortion parameter ξ ranged between 0 and 1 (and slightly more than 1 for certain types of catadioptric camera [Ying and Hu 2004]). Therefore the value of ξ is convenient to bound, interpret and quantize, which are very desirable properties for training CNNs compared to existing polynomial models [Rong et al. 2016]. Notice that a single distortion parameter is generally considered enough to model the distortion [Fitzgibbon 2001].

The unified spherical model relies on a stereographic projection (Figure 1(b)). First, a 3D world point $\mathbf{P}_w = (X, Y, Z)$ is projected onto the sphere at $\mathbf{P}_s = (X_s, Y_s, Z_s) = \mathbf{P}_w / \|\mathbf{P}_w\|$. This spherical point $\mathbf{P}_s$ is then projected onto the image plane at the location $\mathbf{p} = (x, y)$. This projection starts from a point $\mathbf{O}_c$ located at $(0, 0, \xi)$ above the sphere center $\mathbf{O}$. The distance ξ between these two points models the geometric distortion of the camera. The entire projection process can be expressed as:

$$\mathbf{p} = (x, y) = \left(\frac{Xf}{\xi\sqrt{X^2+Y^2+Z^2}+Z} + u_0, \frac{Yf}{\xi\sqrt{X^2+Y^2+Z^2}+Z} + v_0 \right), \quad (1)$$

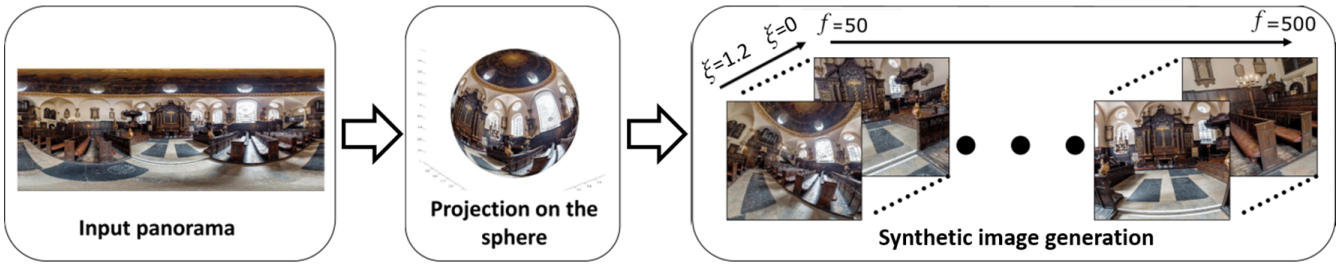


Figure 2: Given an input panorama, we automatically generate images with different focal lengths f and distortion values ξ, via the unified spherical model [Barreto 2006; Mei and Rives 2007][2].

with (u_0,v_0) the pixel coordinates of the principal point in the image, f the focal length (with square pixels), and ξ the distortion parameter [Mei and Rives 2007]. One might notice that when $\xi = 0$ for perspective cameras, Eq. (1) reduces to the standard pinhole perspective projection [Hartley and Zisserman 2004]. As mentioned above, one of the advantages of the spherical model is the closed-form solution of the inverse projection equation [Barreto 2006; Mei and Rives 2007]. Given a 2D image point $\mathbf{p} = (x, y)$, the back-projection from the image to the sphere is computed by:

$$\mathbf{P_s} = (\omega\hat{x}, \omega\hat{y}, \omega - \xi) \text{ with } \omega = \frac{\xi + \sqrt{1 + (1 - \xi^2)(\hat{x}^2 + \hat{y}^2)}}{\hat{x}^2 + \hat{y}^2 + 1}, \quad (2)$$

and

$$\left[\hat{x}, \hat{y}, 1\right]^T \simeq \mathbf{K}^{-1}\mathbf{p} \text{ where } \mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The matrix $\mathbf{K}$ is the standard intrinsic calibration matrix [Hartley and Zisserman 2004]. In this work, we follow the common assumption that the principal point is at the image center, the skew is negligible and the pixel aspect ratio is one (therefore these parameters have been intentionally omitted in the definition of $\mathbf{K}$), and we aim to estimate the focal length f and the distortion parameter ξ.

3.2 Generation of training dataset

In this paper, we investigate a deep learning approach for automatic calibration of wide FOV cameras. To train the deep learning CNN, we need numerous training examples. However, to the best of our knowledge, there is no existing large-scale dataset of wide FOV images with ground truth intrinsic parameters that could be used to train a deep learning network. Concretely, to train our network, we need millions of images with different focal lengths and a large variety of distortions, along with the corresponding ground truth values. In practice, it is cumbersome and virtually impossible to manually capture such data. Furthermore, the calibration of all these cameras would also be hardly feasible for a large-scale dataset. For these reasons, we propose to generate a dataset synthetically.

A straightforward but naive approach would be to generate wide field-of-view images from a set of standard (perspective) calibrated images. However, adding distortion (i.e., increasing the field of view) in standard images inevitably leads to non-visible parts becoming visible, i.e., "occlusion" margins near the image borders typically shown as black area. This leads to non-realistic data. To overcome this limitation, we propose to leverage the large collections of panoramas available on the Internet because their complete 360-degree FOV can emulate any amounts of FOV (see Figure 2). Another advantage of using panoramas is that we can point the virtual camera to different orientations (azimuth and elevation) in order to observe different parts of the scene and mimic tilted cameras, which can provide additional images from a single panorama. In this work, we use panoramas from a dataset collected on the Internet. This dataset contains about 67,000 high-resolution 9104×4552px panoramas acquired in various scenes, such as indoor/outdoor, urban/natural and bright/dark.

Given an input panorama, the generation of a new image with a specific focal length and distortion value is composed of two main steps (see Figure 2). First, the panorama is linearly mapped onto the unit sphere. For this, let us consider a pixel (x, y) in the panorama, and write W and H respectively for the width and the height of the panorama. Then x is converted to the azimuth angle θ such that $x \in (1, W)$ is linearly mapped to $\theta \in (0, 2\pi)$, and similarly y is converted to the elevation angle ϕ such that $y \in (1, H)$ is linearly mapped to $\phi \in (-\pi/2, \pi/2)$. By doing so for each pixel, we can project the panorama onto the sphere. The second step is to create a new synthetic image via a virtual camera, i.e., by reprojection using Eq. (1) with the desired values for focal length f and distortion parameter ξ. In practice, to avoid the artifacts inherent to forward mapping (such as holes, inpainting, and rounding), we instead follow a backward mapping strategy and apply the back-projection of Eq. (2). Some representative results are shown in Figure 2.

By following this approach, we can automatically generate a large-scale dataset composed of millions of images with ground truth intrinsic parameters and covering a large range of image configurations, such as different focal lengths, distortions, appearances, colors, compositions, visible objects, camera types, scene types, etc.

3.3 Network architecture

Given an input image, we follow a deep learning-based approach to predict the distortion parameter and the focal length. We privileged and built upon a state-of-the-art Inception-V3 structure [Szegedy et al. 2016]. We experimented with three different network architectures based on Inception-V3 that we will describe below. For each network architecture, we solved both the classification and regression problems. For the classification problem, we used softmax as the activation function for the output layer(s) and cross entropy for the loss function. For the regression problem, we used the sigmoid activation in the output layer(s) and the logcosh loss.

[2]Credits: panoramic photo by David Iliff with CC-BY-SA 3.0 license (https://creativecommons.org/licenses/by-sa/3.0).

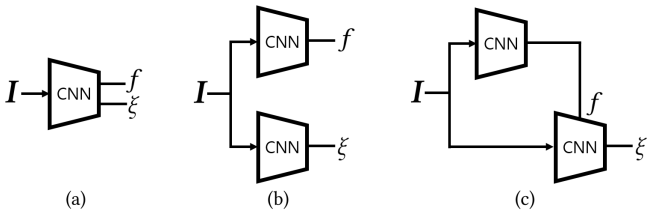


Figure 3: Illustration of the three network architectures: SingleNet (a), DualNet (b) and SeqNet (c). The input is the image I to calibrate, and the outputs are the focal length f and the distortion parameter ξ. In SeqNet, the value of the focal length f estimated by the first network is concatenated into one of the dense layers of the second network.

We now describe the three different network architectures that we experimented with and evaluated. The difference between these three architectures is that they each have specific output layer(s). The first architecture is a single network, that we call SingleNet, with two output dense layers (see Figure 3(a)): one for distortion estimation and one for focal length estimation.

Our second network architecture, called DualNet, is composed of two independent networks (see Figure 3(b)): one outputs the focal length, while the other outputs the distortion value. In other terms, the network in charge of the focal length (resp. distortion) should be invariant to the distortion (resp. focal length).

The third network architecture, called SeqNet, is a sequence of two joint networks (see Figure 3(c)): the first one outputs the focal length, while the second network accepts as an input this estimated focal length value and the input image to estimate the distortion parameter. In practice the value of the focal length estimated by the first network is concatenated into one of the dense layers of the second network.

4 RESULTS

4.1 Parameters of the network

To train the networks listed in Section 3.3, we generated a large-scale dataset composed of millions of images with a resolution of 299×299px from all the panoramic images of the SUN360 dataset (see Section 3.2). For classification training, we used focal lengths on a range between 50 and 500px with a step size of 10, along with distortion values on a range between 0 and 1.2 with a step size of 0.02 (discrete dataset). For regression training, we randomly sampled the values of focal length and distortion parameter on the same ranges described above (continuous dataset). We split the dataset into three subsets: 80% for training, 10% for testing, and 10% for validation. Each original panorama is used exclusively for training, or testing, or validation, i.e., none of the original panoramas belongs to more than one dataset. We performed standard data augmentation by randomly adding Gaussian noise, modifying the brightness and contrast, and image mirroring. Our three networks are pretrained on the ImageNet dataset, and we further train them on our generated dataset with early stop strategy to prevent overfitting [Raskutti et al. 2014]. We set the learning rate to 10^{-5} and used a batch size of 64 for both classification and regression.

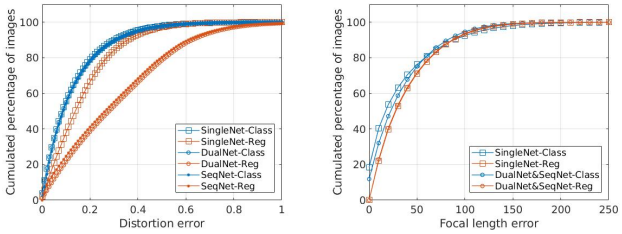


Figure 4: Cumulative error distribution of estimated distortion (left) and focal length (right) with respect to ground truth. In this experiment we compared three network architectures (SingleNet, DualNet and SeqNet), and for each of them both regression (abbreviated Reg) and classification (abbreviated Class).

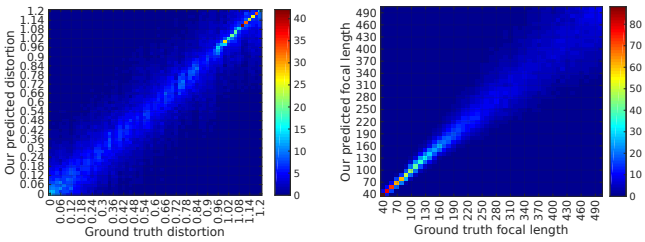


Figure 5: Confusion matrix of our prediction vs. ground truth for the distortion parameter (left) and focal length (right) on the generated image dataset.

4.2 Evaluation

In this section, we evaluate and compare the performance of our three network configurations and regression/classification. The performance is measured on the same continuous dataset. Figure 4 shows the cumulative error distribution for both distortion and focal length with respect to ground truth. First, it shows that, surprisingly, the classification models provide more accurate results than regression models. Second, it shows that the three network architectures (SingleNet, DualNet and SeqNet) have a similar performance for distortion estimation, and SingleNet provides the highest performance, by a small margin, for focal length estimation. DualNet and SeqNet are composed of two networks, while SingleNet is composed of only one network. Therefore the training and feed-forward execution of SingleNet is twice faster than DualNet and SeqNet. That is why SingleNet appears to be the network architecture of choice in terms of accuracy and running time. Therefore we consider only SingleNet-Classification in the remaining of the paper as the architecture with best performance.

Figure 5 shows the confusion matrix of our prediction vs. ground truth for both the distortion and the focal length. It shows that the network consistently estimates the proper values for both variables. Indeed, the plots exhibit a well distinct diagonal. However, we notice a performance decrease when the focal length increases and, inversely, when the distortion parameter decreases. This analysis is consistent with the experiments presented in Section 4.4 and in the supplementary material.

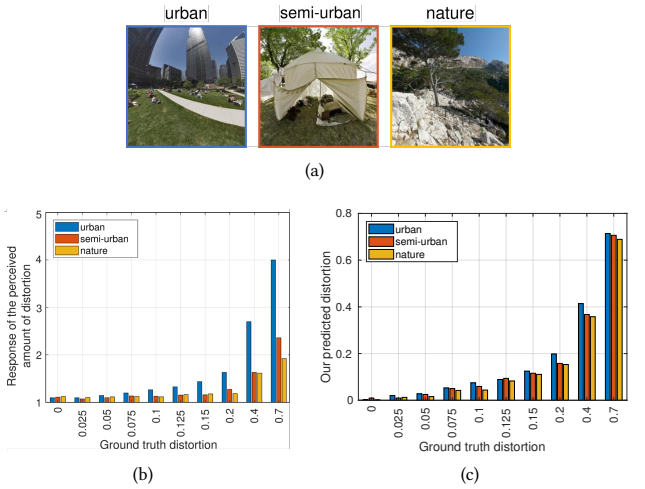


Figure 6: Analysis of the distortion on different scene types. (a) Representative images of the three scene types: urban, semi-urban and nature. (b) User study results on the amount of perceived distortion with respect to the ground truth distortion parameter value. (c) Comparison of our estimated distortion parameter value vs. ground truth.

4.3 User study on human distortion perception

The discrepancy between the values estimated by the network and the ground truth is difficult to assess on a qualitative level from a human perspective. That is why we conducted a user study to measure the required level of accuracy of distortion parameter estimation performed by our network. In this study, we asked participants to rate the amount of perceived distortion in images.

After some preliminary results, we noticed a significant bias on the amount of distortion perceived by the participants for certain types of scene. We debriefed with the participants, and they commented that distortion is clearly visible in images containing lines since they appear as bent curves. In contrast, images with no lines appear less distorted. It is not surprising since lines have been extensively used for wide FOV camera calibration [Antunes et al. 2017; Barreto and Araújo 2005; Bräuer-Burchardt and Voss 2001; Melo et al. 2013; Swaminathan and Nayar 2000; Ying and Hu 2004], and the goal of image undistortion is to make the lines straight.

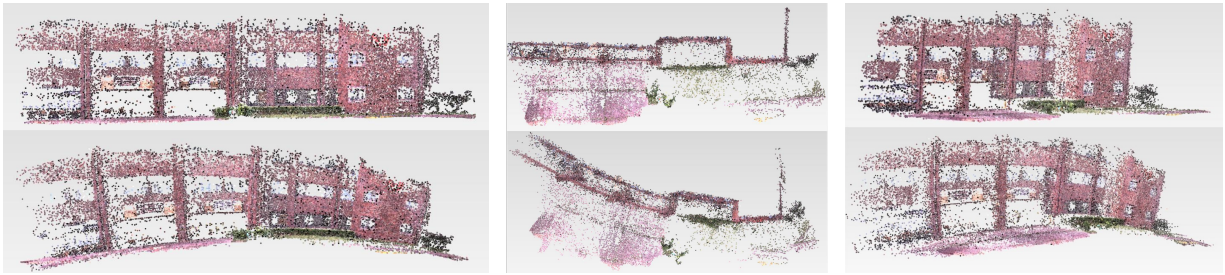
That is why we decided to re-conduct the user study where the images were manually classified into three main scene categories based on the amount of lines: urban (many lines), semi-urban (a very few lines) and nature (no lines at all). Representative images of these categories are available in Figure 6(a). We generated 480 images from 8 different panoramic images per scene category from the 360SUN dataset [Xiao et al. 2012], with focal length set to 150px and different values of the distortion parameter. We sampled the distortion values from 0 to 0.15 with a step of 0.025, plus larger distortions 0.2, 0.4 and 0.7. 14 participants joined the user study and they were asked to answer 'What is the amount of visible fisheye lens distortion?'. They could rate on an integer scale from 1 to 5, where 1 means 'no visible distortion', 2 'very little visible distortion', 3 'moderate amount visible', 4 'strong amount visible', and 5 'very strong amount visible'. Results are available in Figure 6(b). First, it demonstrates that the amount of perceived distortion increases with the actual amount of distortion, as expected. Second, it shows that the amount of perceived distortion indeed depends on the image category: distortions are more visible in urban images than in nature images, even if generated with the actual same value of the distortion parameter. Third, we can use this result to define a success threshold. The figure shows that when the actual distortion is less than 0.2, the amount of perceived distortion for any image categories has a score lower than 2, which corresponds to 'no visible distortion' or 'very little visible distortion'. Therefore, we can set the range of an acceptable distortion error to 0.2. Going back to the evaluation in Figure 4, it means that our success rate for distortion estimation is 78%, i.e., that is the percentage of images whose distortion error is lower than 0.2.

Additionally, we tested the accuracy of our approach on the same set of images that was used in the user study. Figure 6(c) shows our predicted distortion value vs. ground truth for the different scene categories. It shows that our network performs similarly well for the different scene categories, i.e., it deals with urban images (many lines) as well as with natural scenes (no lines at all).

4.4 Comparison to state-of-the-art calibration methods

In this section, we conduct a quantitative comparison to the following popular state-of-the-art calibration methods based on checkerboards: Mei's toolbox [Mei and Rives 2007] (based on the spherical model), OpenCV Brown [Zhang 2000], OpenCV Fisheye [Bradski 2000] (based on the division model [Fitzgibbon 2001]) and Scaramuzza's Toolbox [Scaramuzza et al. 2006]. Additional comparison to line-based approaches [Antunes et al. 2017; Santana-Cedrés et al. 2016] are available in the supplementary material. We experimented with four camera setups. The first three setups consist of a Point-Grey Flea3 camera with a 1328×1048 resolution and equipped with three different lenses: 1) an Avenir 4mm, 2) an Avenir 2.8mm providing a wide FOV, and 3) a Fujinon fisheye lens 1.8mm with a FOV over 180°. The fourth camera setup is a consumer level action camera GoPro HERO6 at a resolution of 1920×1080px. To calibrate the cameras with existing methods based on checkerboards, we acquired about 30 pictures of checkerboard per camera. To calibrate the cameras with our approach, we provided a single image of general scene per camera (see images in the left column of Table 1). To compare the quality of the calibration results, we measure and report the reprojection error. Existing toolboxes directly provide the reprojection error in pixels (since they use checkerboard corners). To compute the reprojection error of our approach, we input our calibration parameters (estimated from a single image of general scene) into Mei's toolbox and performed a pose-only bundle adjustment on the same checkerboard images (i.e., our parameters f and ξ, as well as the principal point (u_0, v_0), are not optimized).

The calibration results are available in Table 1. It contains the estimated focal length f, the set of distortion parameters of their respective camera projection model, as well as the mean reprojection error in pixel. Overall, most of the existing toolboxes obtains a subpixelic accuracy (when used on applicable cameras, as discussed below), and our approach is slightly less accurate than the existing toolboxes. However several important aspects should be



(a) (b) (c)

Figure 7: 3D reconstruction results: front view (a), top view (b) and side view (c). Top row: results obtained with our parameters used for initialization step. Bottom: results obtained with the automatic COLMAP initialization of intrinsic parameters.

Table 1: Camera calibration results for different cameras and calibration methods. All the methods use multiple checkerboard pictures, while ours require a single picture of a general scene.

	Method	f	distortion	mean error (px)
Avenir 2.8mm 	Ours	954	0.77	3.40
	Mei	1973	1.48	0.12
	OpenCV Fisheye	N/A	N/A	N/A
	OpenCV Brown	796	-0.30, 0.17, -0.00, -0.00, -0.07	0.20
	Scaramuzza	788	-788.50, 0, 3.55^{-4}, 2.11^{-7}, -2.62^{-10}	1.01
Avenir 4mm 	Ours	1189	0.47	2.10
	Mei	2270	0.97	0.13
	OpenCV Fisheye	N/A	N/A	N/A
	OpenCV Brown	1158	-0.27, 0.28, 0.00, 0.00, -0.21	0.29
	Scaramuzza	1157	-1157, 0.00, 2.89^{-4}, -2.45^{-7}, 1.79^{-10}	0.54
GoPro 	Ours	1182	0.82	1.11
	Mei	1561	1.28	0.12
	OpenCV Fisheye	787	0.15, -0.77, 1.61, -0.98	0.7
	OpenCV Brown	N/A	N/A	N/A
	Scaramuzza	791	-791.4, 0.00, 0.00, -1.12^{-6}, 3.10^{-10}	0.82
Fisheye 	Ours	777	1.01	1.27
	Mei	1351	1.79	0.10
	OpenCV Fisheye	488	-0.09, 0.65, -1.79, 1.13	1.05
	OpenCV Brown	N/A	N/A	N/A
	Scaramuzza	487	-487.7, 0, 8.18^{-4}, -4.39^{-7}, 4.32^{-10}	1.48

emphasized. First of all, OpenCV Fisheye calibration leads to a reprojection error of more than 100px on Avenir 4mm and Avenir 2.8mm lenses since it is not originally designed for perspective or low-distortion images. These failures cases are noted N/A (for Not Applicable) in the table. In contrast, our approach is applicable on perspective, low-distortion and wide FOV images. Second, similarly, Brown's model is not suitable to handle high distortion [Kannala and Brandt 2006]. In practice, the automatic process in OpenCV Brown skipped around half of the calibration images for the GoPro camera and the Fujinon fisheye lens, which confirms that Brown's model is not adapted to wide FOV cameras. These failures cases are also noted N/A. In contrast, our approach is seamlessly applicable to perspective and wide FOV cameras, and leads to a reprojection error of around 1px for the GoPro camera and the Fujinon fisheye lens. Third, our accuracy remains competitive, for example for wide angle cameras such as GoPro and fisheye lens, especially compared to Scaramuzza's toolbox and the OpenCV Fisheye method. Fourth, we believe that our reprojection accuracy remains overall satisfying, especially considering our assumptions (e.g., principal point at the center of the image and zero skew) and the facts that 1) our approach does not use any calibration target, 2) is based on a single image, 3) does not perform any explicit optimization of the reprojection error, contrary to all these existing methods, and 4) is applied on small 299×299px images, while other methods use high resolution images. Moreover, we will show that our approach can be practically useful for multiple applications, such as initialization for SfM (see Section 4.5) and image undistortion (see Section 4.6).

Finally, we would like to emphasize that our work does *not* aim to compete with checkerboard based approaches in terms of accuracy, but rather aims to provide a solution for camera calibration where traditional techniques cannot be applied, for example single image, images in the wild, or no calibration target.

To conclude this experiment, we would like to underline that calibrating the cameras with a checkerboard took around 30-60 minutes per calibration. This duration includes setting the checkerboard, capturing several images of the checkerboard from different viewpoints, copying the images to the computer, using the calibration toolbox, clicking on corners, selecting a non-radial line (for Mei's toolbox), among other tasks. In contrast, our approach takes less than 2 minutes (to take the picture, copy it to the computer and click on the run button), is fully automatic, needs just a single image of a general scene (i.e., does not require any specific calibration patterns), does not require any setup and the calibration takes around 50ms (on NVIDIA GeForce GTX 1080 Ti GPU).

4.5 3D reconstruction

Camera calibration is an essential step for Structure from Motion (SfM). We conducted experiments where we applied our calibration approach to estimate the intrinsic parameters in order to initialize the SfM pipeline. We used COLMAP [Schönberger and Frahm 2016] along with our bundle adjustment adapted to the unified spherical model, both of which strongly rely on the initialization of the intrinsic parameters.

For this experiment, we used an Avenir 4mm lens mounted on a PointGrey Flea3 camera, and acquired an image sequence composed of around 300 images. We applied our calibration approach on the images of the sequence, and used the median values of the focal length and distortion parameter for SfM initialization. A representative 3D reconstruction result is shown in Figure 7, and additional results are available in the supplementary material. It demonstrates the drastic difference between the results obtained using the camera parameters estimated by our approach (top row) and by the automatic COLMAP parameter initialization without EXIF file (bottom row). Notice that in both cases, the initial parameters are refined during the bundle adjustment. In Figure 7(a)-bottom, the reconstruction by the COLMAP initialization has a clearly visible

Figure 8: Examples of automatic undistortion results on images in the wild[3]. Left: Original image. Right: output of our algorithm. One may note the great variety of image appearance, including close/far objects, indoor/outdoor, true color/photoshopped, vertical/tilted viewpoint, and ground/aerial views.

curved horizon. Similarly, in the top view of Figure 7(b)-bottom, we can observe that the wall edges are bent. Moreover, in most cases, the reconstruction cannot be operated without these initial parameters. In contrast, Figure 7-top shows that our approach can correctly preserve the geometric structures of the building, such as the straight walls, parallel windows and horizon.

4.6 Undistortion of images in the wild

To challenge the robustness of our algorithm, we propose to undistort a set of wide FOV images "in the wild", i.e., downloaded from the Internet (without ground truth available). The images have been captured from unknown types of cameras and lenses having different characteristics, such as camera model, lens type, focal length, distortion, optics quality, resolution and light sensitivity. Given an image, we apply our network to estimate the focal length and distortion parameter. Then for the undistortion itself, the input distorted image is back-projected on the unit sphere using Eq. (2) and the estimated intrinsic parameters. This spherical image is then projected on the image plane with the desired intrinsic parameters. For instance, to generate a perspective image exempt of any distortions (pinhole model), we fixed ξ to 0 and the focal length to 150px. Note that other focal length values could be used for different zooming and cropping effects when ξ is set to 0.

A set of representative results is available in Figure 8. It shows that our algorithm is able to correctly predict the distortion parameter and focal length under various scenarios and environments. For instance, our network is able to deal with indoor (Figure 8(f,h,m,p)) and outdoor (Figure 8(b,c,e,r)) scenes, as well as under different

[3]Credits: (a) photo by Pau Brown with license CC-BY 2.0; (b) photo by Stig Nygaard with license CC-BY 2.0; (c) photo by Nan Palmero with license CC-BY 2.0; (d) photo by Eli Christman with license CC-BY 2.0; (e) photo by Eli Christman with license CC-BY 2.0; (f) photo by Chris Dag with license CC-BY 2.0; (g) photo by Robert Couse-Baker with license CC-BY 2.0; (h) photo by Matthew Paulson with license CC-BY-NC-ND 2.0 (https://creativecommons.org/licenses/by-nc-nd/2.0/); (i) photo by Paul Stevenson with license CC-BY 2.0; (j) photo by Jake Cook with license CC-BY 2.0; (k) photo by Raissa Ruschel with license CC-BY 2.0; (l) photo by Yann Gar with license CC-BY 2.0; (m) photo by Chris Dag with license CC-BY 2.0; (n) photo by Michael Guthmann with license CC-BY 2.0 (https://creativecommons.org/licenses/by/2.0/); images (o), (p), (q), (r) were acquired by the authors.

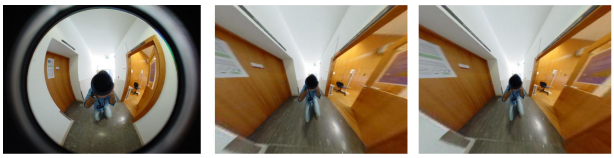


Figure 9: Comparison of catadioptric camera calibration. Left: input catadioptric image[4]. Middle: undistortion result by the state-of-the-art Barreto's toolbox [Barreto and Araújo 2005] with manual selection of lines. Right: undistortion result by our automatic deep learning approach.

lighting conditions like night (see Figure 8(k) and day (Figure 8(b)). Our approach can handle images modified with a heavy image editing process, such as HDR color manipulation (Figure 8(j)). Our network shows also robustness to "confusing" images, such as Figure 8(g) with a curved reflective surface. Similarly, clutter environment like Figure 8(f) can be handled by our approach. In addition, our approach also handles various camera viewpoints like tilt upward (Figure 8(d)), close objects (Figure 8(i,q)) and strong perspective effect (Figure 8(a,l,n)). Moreover, it can also handle images with naturally distorted contents – not to be undistorted – such as circular patterns (Figure 8(j)). Furthermore, under challenging conditions, like in Figure 8(q), where a face covers most of the image, our network also leads to visually satisfying results. Finally, our algorithm can successfully handle images acquired from a drone camera (Figure 8(o)) and a smartphone with clip-on fisheye lens (Figure 8(p)). Overall, this set of results demonstrates the adaptability of our approach to process images with various appearances and characteristics. Additional results are available in the supplementary material.

In addition to these experiments with dioptric cameras, we also tested our approach on the challenging case of catadioptric cameras. A representative example is available in Figure 9. In this figure, a catadioptric image has been automatically and successfully undistorted by our approach. For comparison, we also provide the result obtained by Barreto's calibration toolbox dedicated to catadioptric cameras where the user has to manually extract long lines [Barreto and Araújo 2005]. It shows that our result is visually as satisfying as the one obtained by manual state-of-the-art calibration with lines.

4.7 Limitations and future work

While our approach is accurate and versatile (as shown in the above experiments), it still has limitations. For example, we observed that our approach has difficulties with challenging images affected by strong motion blur and over-exposure. A representative example is shown in Figure 10(a), where the network returns inaccurate intrinsic parameters leading to an unsatisfying image undistortion (see the convex lines at the bottom of the wall in the undistorted image). Note that strong motion blur and over-exposure are also a limitation common to the existing calibration methods.

Another limitation is the rolling shutter effect. An example is shown in Figure 10(b), where the input image contains rolling shutter artifacts [Zhuang et al. 2017]. Training a network to determine

[4]Credits: photo by João P. Barreto [Barreto and Araújo 2005]

(a) (b)

Figure 10: Representative failure examples: (a) input image affected by strong motion blur and over-exposure (left), and the resulting undistortion (right), (b) input image[5] affected by rolling shutter (left) and the resulting undistortion (right).

if the curvatures present in an image are induced by the lens distortion or rolling shutter is an interesting direction for future work.

We also observed that our intrinsic estimation tends to be less accurate for low distortion (see Figure 5), even if the network was trained on this type of images. We believe that it might be due to the uniform distribution of distortion and focal length parameters during the data generation. In practice, the "amount" of visually perceived distortion in images seems to increase with ξ very quickly in a non-linear manner (see Figure 6(b)). This may lead to an unbalanced data generation in favor of cameras equipped with wide angle lenses. That is why a promising direction for future work would be the generation of images using a parameter distribution adapted to the quantity of perceived distortion in the image.

The unified spherical model and other existing lens distortion models (see Section 3.1) have an ambiguity between the focal length and distortion parameter [Cornelis et al. 2002; Hartley and Kang 2007; Li and Hartley 2005], i.e., different combinations of focal length and distortion values may lead to the same (or similar) projection of a world point in the image. Therefore a promising extension of our approach is to incorporate the mathematical relationship between the focal length and distortion into the loss function.

5 CONCLUSION

We have presented the first deep learning-based approach for automatic intrinsic calibration of wide FOV cameras. The only required input is a single image of a general scene, and our approach can automatically estimate the focal length and distortion parameter. For this, we introduced a method to automatically generate a large-scale dataset of wide FOV images with ground truth intrinsic parameters in order to train the CNN. We also investigated three different network architectures and observed that SingleNet is the network of choice in terms of accuracy and execution time (both training and running time). We have demonstrated the accuracy of our approach in various experiments on synthetic data and real images. Additionally, we successfully applied our approach on various cameras, such as machine vision cameras equipped with fisheye lens, GoPro cameras, smartphones with clip-on fisheye lens, and catadioptric cameras, demonstrating the robustness of our approach and its general applicability. Moreover, experiments also demonstrated that our approach can correctly handle wide FOV images "in the wild". We also compared our results to several state-of-the-art calibration methods and showed that a great advantage of our approach is that it can be successfully used for several practical cases when

[5]Credits: photo by Bingbing Zhuang [Zhuang et al. 2017]

existing state-of-the-art methods cannot be applied, for example single image, images in the wild, unstructured scenes, absence of lines, no calibration target and/or manual process. Finally, our work constitutes a first large-scale benchmark and provides evaluation dataset for future research on camera calibration. Our code and dataset are available on our project website.

ACKNOWLEDGMENTS

This research was partially supported by the KAIST High Risk High Return Project (HRHRP) and KAIST Research Promotion Team through URP program. It was also partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2017R1C1B5077030). F. Rameau was supported by Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2015H1D3A1066564). We are very grateful to Michel Antunes and João P. Barreto for running their line-based method on our image dataset. We also thank Amirsaman Ashtari for the drone image acquisition.

REFERENCES

Michel Antunes, João P. Barreto, Djamila Aouada, and Björn Ottersten. 2017. Unsupervised Vanishing Point Detection and Camera Calibration from a Single Manhattan Image with Radial Distortion. In *CVPR*.

João P. Barreto. 2006. A Unifying Geometric Representation for Central Projection Systems. *CVIU* (2006).

João P. Barreto and Helder Araújo. 2005. Geometric Properties of Central Catadioptric Line Images and Their Application in Calibration. *TPAMI* (2005).

Jean-Charles Bazin, Cédric Demonceaux, Pascal Vasseur, and In So Kweon. 2010. Motion Estimation by Decoupling Rotation and Translation in Catadioptric Vision. *CVIU* (2010).

Jean-Charles Bazin, Cédric Demonceaux, Pascal Vasseur, and In So Kweon. 2012. Rotation Estimation and Vanishing Point Extraction by Omnidirectional Vision in Urban Environment. *IJRR* (2012).

Sean Bell, Kavita Bala, and Noah Snavely. 2014. Intrinsic Images in the Wild. *TOG* (2014).

Gary Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

Christian Bräuer-Burchardt and Klaus Voss. 2001. A New Algorithm to Correct Fish-Eye-and Strong Wide-Angle-Lens-Distortion From Single Images. In *ICIP*.

Duane C. Brown. 1971. Close-Range Camera Calibration. *Photogrammetric Engineering* (1971).

Robert Carroll, Maneesh Agrawala, and Aseem Agarwala. 2009. Optimizing Content-Preserving Projections for Wide-Angle Images. *TOG (SIGGRAPH)* (2009).

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-Image Depth Perception in the Wild. In *NIPS*.

Kurt Cornelis, Marc Pollefeys, and Luc Van Gool. 2002. Lens Distortion Recovery for Accurate Sequential Structure and Motion Recovery. In *ECCV*.

Fisheye-Hemi. 2015. https://imadio.com/products/prodpage_hemi.aspx. (2015).

Andrew Fitzgibbon. 2001. Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion. In *CVPR*.

Simone Gasparini, Peter Sturm, and João P. Barreto. 2009. Plane-Based Calibration of Central Catadioptric Cameras. In *ICCV*.

Christian Häne, Lionel Heng, Gim Hee Lee, Alexey Sizov, and Marc Pollefeys. 2014. Real-Time Direct Dense Matching on Fisheye Images Using Plane-Sweeping Stereo. In *3DV*.

Richard Hartley and Sing Bing Kang. 2007. Parameter-Free Radial Distortion Correction with Center of Distortion Estimation. (2007).

Richard Hartley and Andrew Zisserman. 2004. *Multiple View Geometry in Computer Vision.* Cambridge University Press.

Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matt Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. 2018. A Perceptual Measure for Deep Single Image Camera Calibration. In *CVPR*.

Ciaran Hughes, Patrick Denny, Martin Glavin, and Edward Jones. 2010. Equidistant Fish-Eye Calibration and Rectification by Vanishing Point Extraction. *TPAMI* (2010).

Fangyuan Jiang, Yubin Kuang, Jan Erik Solem, and Kalle Åström. 2014. A Minimal Solution to Relative Pose with Unknown Focal Length and Radial Distortion. In *ACCV*.

Sing Bing Kang. 2000. Catadioptric Self-Calibration. In *CVPR*.

Juho Kannala and Sami S. Brandt. 2006. A Generic Camera Model and Calibration Method for Conventional , Wide-Angle, and Fish-Eye Lenses. *TPAMI* (2006).

Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. 2013. Motion Estimation for Self-Driving Cars with a Generalized Camera. In *CVPR*.

Hongdong Li and Richard Hartley. 2005. A Non-Iterative Method for Correcting Lens Distortion from Nine Point Correspondences. *OMNIVIS* (2005).

Wen-Yan Lin, Linlin Liu, Yasuyuki Matsushita, Kok-Lim Low, and Siying Liu. 2012. Aligning Images in the Wild. In *CVPR*.

Peidong Liu, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. 2017. Direct Visual Odometry for a Fisheye-Stereo Camera. In *IROS*.

Christopher Mei and Patrick Rives. 2007. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *ICRA*.

Rui Melo, Michel Antunes, João P. Barreto, Gabriel Falcão, and Nuno Gonçalves. 2013. Unsupervised Intrinsic Calibration from a Single Frame Using a "Plumb-Line" Approach. In *ICCV*.

Márcio Mendonça, Ivan N. Da Silva, and José E.C. Castanho. 2002. Camera Calibration Using Neural Networks. *WSCG* (2002).

Branislav Micusík and Tomás Pajdla. 2003. Estimation of Omnidirectional Camera Model from Epipolar Geometry. In *CVPR*.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. 2014. Early Stopping and Non-Parametric Regression: an Optimal Data-Dependent Stopping Rule. *JMLR* (2014).

Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016. EgoCap: Egocentric Marker-less Motion Capture with two Fisheye cameras. *TOG (SIGGRAPH Asia)* (2016).

Jiangpeng Rong, Shiyao Huang, Zeyu Shang, and Xianghua Ying. 2016. Radial Lens Distortion Correction Using Convolutional Neural Networks Trained with Synthesized Images. In *ACCV*.

Daniel Santana-Cedrés, Luis Gomez, Miguel Alemán-Flores, Agustín Salgado, Julio Esclarín, Luis Mazorra, and Luis Alvarez. 2016. An Iterative Optimization Algorithm for Lens Distortion Correction Using Two-Parameter Models. *Image Processing On Line* (2016).

Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. 2006. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *IROS*.

Johannes L. Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *CVPR*.

Thomas Schöps, Torsten Sattler, Christian Häne, and Marc Pollefeys. 2017. Large-scale Outdoor 3D Reconstruction on a Mobile Device. *CVIU* (2017).

Shishir Shah and J. K. Aggarwal. 1994. A Simple Calibration Procedure for Fish-Eye (High-Distortion) Lens Camera. In *ICRA*.

Birger Streckel, Jan-Friso Evers-Senne, and Reinhard Koch. 2005. Lens Model Selection for a Markerless AR Tracking System. In *ISMAR*.

Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João P. Barreto. 2011. Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Foundations and Trends in Computer Graphics and Vision* (2011).

Rahul Swaminathan and Shree K. Nayar. 2000. Nonmetric Calibration of Wide-Angle Lenses and Polycameras. *TPAMI* (2000).

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR*.

Zhongwei Tang, Rafael Grompone von Gioi, Pascal Monasse, and Jean-Michel Morel. 2017. A Precision Analysis of Camera Distortion Models. *TIP* (2017).

Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. 2015. DeepFocal: a Method for Direct Focal Length Estimation. In *ICIP*.

Scott Workman, Menghua Zhai, and Nathan Jacobs. 2016. Horizon Lines in the Wild. In *BMVC*.

Jianxiong Xiao, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2012. Recognizing Scene Viewpoint Using Panoramic Place Representation. In *CVPR*.

Yalin Xiong and Kenneth Turkowski. 1997. Creating Image-Based VR Using a Self-Calibrating Fisheye Lens. In *CVPR*.

Xianghua Ying and Zhanyi Hu. 2004. Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model. In *ECCV*.

Xianghua Ying and Hongbin Zha. 2008. Identical Projective Geometric Properties of Central Catadioptric Line Images and Sphere Images with Applications to Calibration. *IJCV* (2008).

Mi Zhang, Jian Yao, Menghan Xia, Kai Li, Yi Zhang, and Yaping Liu. 2015. Line-Based Multi-Label Energy Optimization for Fisheye Image Rectification and Calibration. In *CVPR*.

Zhengyou Zhang. 1996. On the Epipolar Geometry Between Two Images with Lens Distortion. In *ICPR*.

Zhengyou Zhang. 2000. A Flexible New Technique for Camera Calibration. *TPAMI* (2000).

Bingbing Zhuang, Loong-Fah Cheong, and Gim Hee Lee. 2017. Rolling-Shutter-Aware Differential SfM and Image Rectification. In *ICCV*.