Learning to Rectify Fisheye Images Under Geometric Constraints

Zhu-Cun Xue, Nan Xue, Jiangning zhang, Gui-Song Xia, *Senior Member, IEEE*

Abstract—This work addresses the problem of correcting fisheye distortion by utilizing implicit geometric constraints. In this paper, assuming that the distorted lines generated by fisheye projection should be straight after rectification, we propose a promising exploration of applying the geometric constraints into the distortion rectification task, and present a novel fisheye image rectification network (FirNet) to simultaneously calibrate distortion parameters of fisheye lens and rectify the distorted images. Considering the nonlinearity of distortion distribution in fisheye image, a multi-scale calibration module is designed to compensate the rectification effect on the whole image. In addition, we also design a attention selection module to solve the inaccurate estimation of distorted lines by optimizing the uncertain pixel-errors. To better train and evaluate the proposed model, we also create a new large-scale dataset labeled with corresponding distortion parameters and well-annotated distorted lines. Compared with the state-of-the-art methods, our model achieves the best published rectification quality and the most accurate estimation of distortion parameters on a large set of real and synthetic fisheye images.

Index Terms—Deep learning, Fisheye image rectification, Distortion parameters estimation, Geometric guidance

1 Introduction

Fisheye cameras enable us to perceive the world visually with ultra-large field of view (FoV) and have been widely used in many vision-based applications such as automatic driving [1], virtual reality [2], video surveillance [3] and panoramic machine vision navigation [4], to obtain more visual information of the scene in a single image or video frame. However, the captured fisheye images often suffer from severe geometric distortions, as the large FoV of the camera is usually achieved by applying a non-linear mapping function (*e.g.*, stereographic, equidistant, or orthographic) to the imaging systems. Thus, before developing a vision system equipped with fisheye cameras, it is always required to calibrate the parameters of the cameras and eliminate the image distortions, especially considering the fact that most of current computer vision algorithms (*e.g.*, Structure-from-Motion and SLAM) strictly rely on the assumption of pinhole camera model.

Early works used specific 2D/3D calibration patterns associated with salient features (*e.g.* checkerboard), and formulated the camera calibration as an optimization problem by fitting a pinhole camera model from the correspondences between the control points and their projections in images [5]–[7]. This type of methods well handle the case that the imaging processes strictly obey to the pinhole camera model, but always failed to deal with the nonlinear geometrical distortions in wide-angle and fisheye images. Later, the model as well as the calibration methods were extended to more general imaging models [8]–[11]. Despite promising results have been achieved by such methods, they usually require well-prepared calibration patterns and extra laborious manual operations to measure the 3D positions of control points for precisely establishing the 2D-3D correspondences, which seriously limit their usage scenarios in real applications.

Instead of relating the control points between multiple images, subsequent investigations proposed to detect geometric objects (*e.g.*, conics, straight lines or spheres) from a single image and further exploit their correspondences in 3D world [5], [12]–[17]. Specifically, some of them impose that straight lines or spherical objects in 3D scene should appear straight or circular in 2D image, if there is no any distortion effect for the camera lenses. These calibration approaches are more flexible to handle the calibration problem in different application scenarios. However, they tend to be uncontrollable when specified geometric features cannot be accurately detected, and the involved detection of geometric objects in fisheye images itself is another challenging problem in computer vision.

Recently, the advances of deep learning motivated the community to renew the problem of fisheye lens calibration by utilizing learned deep features [18], [19]. Instead of explicitly detecting the geometric objects in fisheye images, these methods tried to learn discriminative image features with convolutional neural networks (CNNs) and fit a function from the space of distorted images to the space of camera parameters in an end-to-end manner. Consequently, the difficulties of detecting geometric objects can be alleviated by learning deep features of the scene, and state-of-the-art performances on fisheye image rectifications were reported. While, it is worth noticing that the geometrical characteristics have not been fully exploited in the calibration process by using CNNs, and the aforementioned methods often fail in real-world scenarios with more severe geometrical distortions due to the issue of overfitting.

When using deep models to correct the serve distortions in fisheye images, we should observe that the explicit scene geometries are still strong constrainss [12], [14], [17] to boost the rectification performance. Specifically, as a kind

All the authors are with the Department of Computer Science and the State Key Lab. LIESMARS, Wuhan University, Wuhan, 430079, China. e-mail: {zhucun.xue, xuenan, guisong.xia}*@whu.edu.cn*

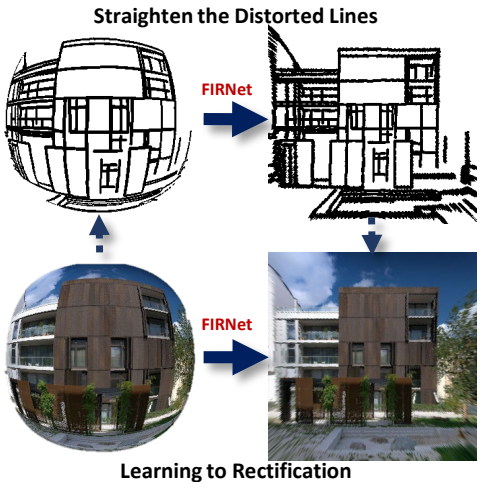


Fig. 1. Implicit geometric constraint in fisheye images: the distorted lines generated by fisheye projection should be straight in normal perspective images. How to apply this strong constraint which is widely used in traditional calibration into an end-to-end rectification learning is the main problem to be solved in this paper.

of fundamental geometric primitive, straight lines are very common to see at the object boundaries of natural images. Thus, it is of great interest to exploit a method to apply such fundamental geometric property of straight lines for the pinhole camera model (*i.e., the projection of a straight line from space to the camera should be a line* [12]) to fisheye image calibration deep networks.

In this paper, we propose a novel *Fisheye Image Rectification Network* (FIRNet), which combines the advantages of the reliable geometrical theory [12] and the strong discriminative power of CNNs, to learn distortion parameters by exploiting the explicit scene geometries. As shown in Fig. 1, the *distorted lines* that generated by fisheye projection could characterize the distortion effect of input images implicitly, and if FIRNet could straighten the distorted lines, the fisheye distortion could be well rectified meanwhile.

1.1 Method Overview

In order to take advantage of the geometric information in fisheye images, *Fisheye Image Rectification Network* (FIRNet) is designed to rectify fisheye distortion under the constraint of straightening the detected distorted lines. FIRNet consists of two main components: the *Line-aware Rectification (LR)* module and the *Attention-based Refinement (AR)* module, as shown in Fig. 2.

Line-aware rectification (LR) module. Given a fisheye image as input, the LR module aims at learning to straighten the detected distorted lines which are distorted due to fisheye effect, and in this process, the distortion parameters of the fisheye image and the corresponding corrected image are output, as shown in the bottom of Fig. 2. In general, LR module goes through the following processes in turn: (1) First, in order to extract these distorted lines with minimized noise interference efficiently, a *distorted line perception* (DLP) sub-module inspired by [20] is designed to address these difficulties. Believing that these geometric information could improve the performance of rectification, and then we feed these distorted lines map together with RGB fisheye image into the following rectification network. (2) Besides, a *multi-scale calibration* sub-module is proposed to eliminate the non-linearity of distortion distribution in fisheye image, by perceiving both the local and the global features respectively. (3) At last, a *rectification* sub-module rectifies the input fisheye images as well as the distorted lines map by using the learned parameters.

Attention-based Refinement (AR) module. Although the LR module can achieve promising rectification performance, it still suffers from the detection errors of distorted lines that caused by the DLP sub-module. In order to reduce its impact on the network and obtain more refined rectification results, we design the attention-based refinement module to learn which pixel is most likely to have uncertain errors and weaken these uncertainty-guided influence. As shown in the top of Fig. 2, the original fisheye image, the features from middle layers as well as the rectified image from LR module are all input into the AR module, and an uncertain map is obtained finally by calculating the probabilities distribution of each pixel, which helps the network to achieve a more robust optimization and get better distortion parameters.

In summary, LR module is able to accomplish the fundamental rectification work, while the AR module helps the LR module achieve more refined results. As seen in Fig. 2, for any single fisheye image input into FIRNet, LR and AR modules complement each other to achieve the task of online rectification in real-time, and also output the rectified images as well as the distortion parameters. In order to train FIRNet, it is required to feed ground truth of the rectified images without distortion as well as the precisely measured distortion parameters into the network as supervisions. However, to the best of our knowledge, there is no such a dataset simultaneously contains these different kinds of data samples. Thus, we create a new large-scale dataset with accurately labeled distortion parameters as well as the labeling of distorted lines, specially for the fisheye lens calibration. This dataset consists of two main parts: *D-Wireframe* which converts the wireframe dataset [20] to distorted wireframe collections by randomly given distortion parameters; *Fish-SUNCG* which utilizes the 3D model repository of SUNCG [21] to render the imaging of real fisheye lens in 3D virtual scenes. With the help of *D-Wireframe* and *Fish-SUNCG*, FIRNet could achieve excellent rectification performance end-to-end and also correct the wild fisheye images download from Internet well.

1.2 Our Contributions

Our contributions can be summarized as follows:

- A new exploration to impose explicit geometry constraints onto the process of learning to rectify fisheye distortion.
- An end-to-end network which (1) balances the nonlinear distribution of distortion effects by multi-scale perception, (2) calculates the geometric errors with emphasis on the distorted lines, (3) and concerns the pixel uncertainty to implement more refined rectification results.
- A large scale fisheye dataset with accurate distortion parameter as well as complex geometric labeling.

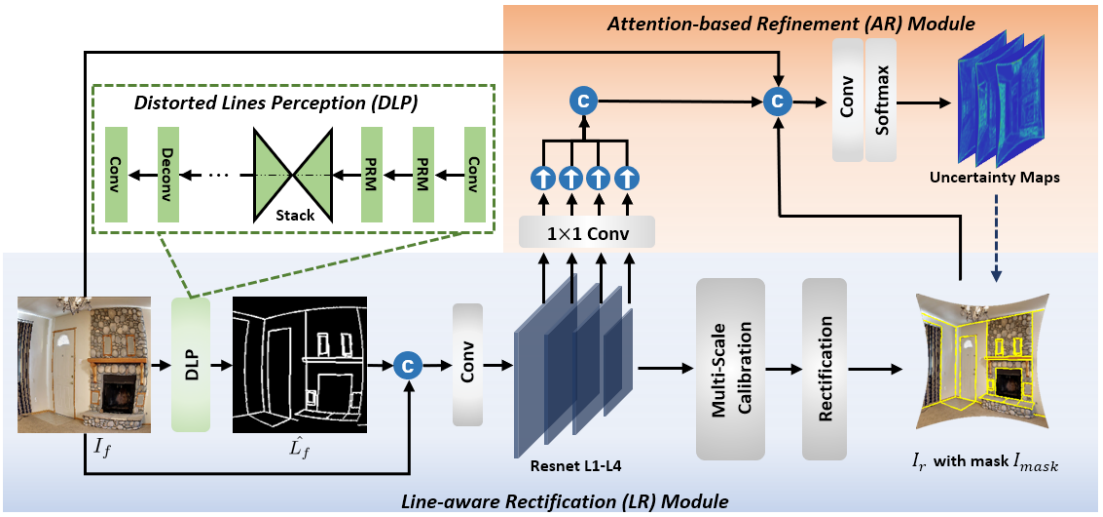


Fig. 2. Overview of the proposed FirNet. The whole network architecture consists of two modules: the Line-aware Rectification (LR) module and the Attention-based Refinement (AR) module. The former estimates the distortion parameters by learning how to straighten the distorted lines, and it contains the distorted lines perception sub-module (DLP), a multi-scale calibration sub-module as well as a rectification sub-module; and the latter focuses on the uncertainty of pixels for achieving a more robust optimization. The symbols Ⓒ and Ⓣ denote concatenation and up-sampling operation respectively.

A preliminary version of this paper has been published [22]. In this paper, we further improve the performance of fisheye calibration in the following aspects:

- For the distorted line detection, we introduce an attention-based refinement module to estimate the uncertainty of the line maps and then propose an uncertainty-guided pixel loss to learn better distorted line maps.
- To further exploit the implicit geometrical constraint in fisheye images, we improve the geometric loss by adding a global weight mask, which allows the network focuses on the structural distorted lines when calculating the global pixel loss.
- The synthesized dataset is enlarged with more variety distortion parameters, for fitting the distribution of the changeable fisheye images in the real world as well as possible.

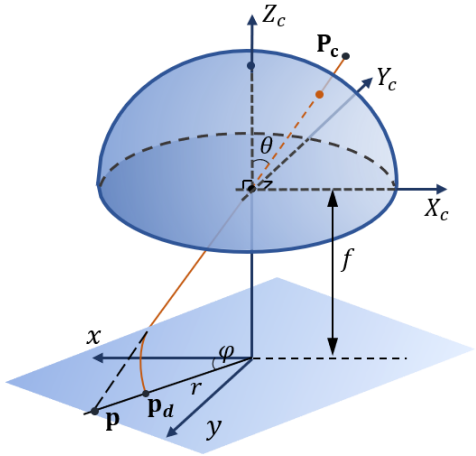


Fig. 3. Camera projection. $\mathbf{p}$ and $\mathbf{p}_d$ are the projection points of the point $\mathbf{P}_c$ through pinhole and fisheye lens respectively.

2 Background and Related Work

2.1 Fisheye Projection Model

For pinhole camera, its projection follows the principle of perspective imaging. Given a normal pinhole camera with focal length f, the perspective projection model can be described as $r = f\tan\theta$, where r indicates the projection distance between the principal point and the point in the image, and θ is the angle between the incident ray and the camera's optical axis. However, fisheye lens violates this perspective projection model [23], [24] thus cause nonlinear distortion of light during imaging, which is completely different against the pinhole camera, as shown in Fig. 3. Since the images captured by the fisheye lens follow various projections, the existing projection models are not applicable, such as stereographic, equidistance, equisolid angle and orthogonal. To solve above problem, the general polynomial projection model proposed by [10] is often used to automatic approximate the imaging rules, which is suitable for real fisheye lens conventional and wide-angle lens. The model can be represented by a formula:

$$r(\theta) = \sum_{i=1}^{n} k_i \theta^{2i-1}, \quad n = 1, 2, 3, 4, \ldots \tag{1}$$

Generally, this model can accurately estimate the image formation of fisheye lens when n reaches five [10]. Therefore, we take the sum of the first five polynomial terms as our final fisheye projection model in this paper. As shown in Fig. 3, given a 3D scene point $\mathbf{P}_c := (x_c, y_c, z_c)^T \in \mathbb{R}^3$ in the camera coordinate system, it will be projected into $\mathbf{p}_d := (x_d, y_d)^T \in \mathbb{R}^2$ in the image plane through the refraction of the fisheye lens and $\mathbf{p} := (x, y)^T \in \mathbb{R}^2$ through pinhole lens without distortion. Since the parameter θ is shared between above two projection models, the correspondence between $\mathbf{p}_d$ and $\mathbf{p}$ can be expressed as, $\mathbf{p}_d = r(\arctan(r/f))(\cos\varphi, \sin\varphi)^T$, with $\varphi = \arctan((y_d - y)/(x_d - x))$ indicating the angle

between the ray which connects the projected point and the center of image and the x-axis of the image coordinate system. Assuming that the pixel coordinate system is orthogonal, we can get pixel coordinate $P_f(u, v)$ converted from image $\mathbf{p}_d$ as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} m_u & 0 \\ 0 & m_v \end{bmatrix} \begin{pmatrix} x_d \\ y_d \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{2}$$

The principal point of fisheye image is represented as (u_0, v_0), as m_u and m_v indicate the number of pixels per unit distance in horizontal and vertical direction respectively.

By using Eq. (2), the distortion effect of fisheye images can be rectified once we obtain the parameters $K_d = (k_1, k_2, k_3, k_4, k_5, m_u, m_v, u_0, v_0)$. Thus, we will accurately estimate the parameters K_d for any given fisheye image and eliminate the image distortion simultaneously in this paper. And the fisheye projection model is also used to generate datasets for training and testing.

2.2 Related Work

Fisheye Distortion Rectification. Solving the distortion caused by fisheye lens is still the main problem in camera calibration, and many researches have been devoting themselves to fisheye calibration and distortion correction studies in the past decades. The traditional approaches which depend on the view geometry can be broadly divided into two categories. One is the marker-based calibration which usually requires known patterns or objects in the specific calibration field, thus it can estimate the relationship between the detected pattern points in the 3D space and their projections on the 2D image plane to obtain the distortion parameters. [8]–[10], [25], [26] use 2D pattern calibration board of known geometry (e.g., checkerboard or dot-array plate) to achieve distortion rectification through the designed project model and the prior of known pattern positions, while [27] adopt the calibration method based on 3D pattern which estimates the projection matrix of 3D-2D correspondence. In this paper, we mainly refer to the fisheye projection model in [10] which adopts polynomial to fit the function of lens projection, so any type of lens can be stably calibrated. However, although the marker-based calibration methods can achieve high-precision calculation results, it is costly to build such calibration fields of large-scale and high-precision as well as it usually turns to be laborious and time-consuming for manually labeling every calibration pattern.

By contrast, another calibration method relies on the structural information of the single fisheye image itself or the motion of multiple images, rather than the known geometric pattern. Certain methods [12], [14], [15], [17], [28] assume that the extracted lines or arcs in fisheye images should to be straight after rectification. Devernay *et al.* [12] proposes that the straight line segments in the real world should maintain their line property after the projection of fisheye lens. Along this axis, Bukhari *et al.* [15] recently uses an extended Hough Transform of lines to correct radial distortions. And the 'plumb-lines' are used to rectify fisheye distortions [14], [16], [17] with a similar assumption. Although these type of calibration methods are simple and efficient, the quality of final correction result is unsatisfactory due to the accuracy of geometric structure extraction and the difficulty of optimization calculation.

With the recent development of the deep learning, some researches add representational features extra by CNNs to the processes of fisheye distortion rectification [18], [19], which could mitigate the difficulty of detecting geometric objects in distorted images. [18] is the first one using CNNs for radial distortion correction, which regards the radial distortion rectification as a classification problem of predicting 401 distortion coefficients. However, because of the simple structure and rigid framework, this method is difficult to deal with distorted images of various types. Based on the above-mentioned drawbacks, the FishEyeRecNet [19] proposes an end-to-end model that introduces a semantic scene parsing module into the rectification of fisheye images and has reported promising results. But it is not clear what kind of high-level geometric information the model learns, where the intelligibility of the network is important for fisheye image rectification. To better encode the explicit geometry [14], [16], [17] in fisheye images with CNNs to assist distortion correction effectively, we propose a new exploration: correction of fisheye distortion in the learning to straighten *distorted lines* which generated by fisheye projection.

Attention Selection. Inspired by human attention mechanism theory [29], attention mechanism allows us to focus on the most relevant and important parts of the image. Mnih *et al.* [30] presents a novel recurrent neural network model with a hard alignment, and it could adaptively focus on the key regions in images and locations in videos. Subsequently, Bahdanau *et al.* [31] proposes an attention model with differentiable soft alignments for machine translation, which greatly boosts the translation performance. Besides, the attention selection mechanism gradually applies to image generation tasks. Gregor *et al.* [32] introduces the DRAW neural network architecture combined with a novel spatial attention mechanism, which enables the generator automatically focuses more on important objects of the input image. SAGan [33] allows attention-driven remote dependency modeling for image generation tasks and AttentionGAN [34] also uses an additional attention network in object transfiguration, where both of them achieve satisfying results. Inspired by these works, we propose an attention-based refinement module to learn the pixel-guided distortion effects, that is, the pixel uncertain error still exists in rectified image, which can significantly boost the quality of the final outputs as well as lead to a more robust optimization result in distortion rectification task.

3 Fisheye Rectification Network

In this section, we propose a novel fisheye rectification network - FirNet, to carry out fisheye distortion rectification problem with the help of geometric guidance. It aims at estimating distortion parameters as well as learning to straighten the distorted lines simultaneously, and the overall network architecture is depicted in Fig. 2. For any given RGB fisheye image I_f, the *Line-aware Rectification (LR)* module firstly detect the distorted lines $\hat{L_f}$ in I_f as preparatory

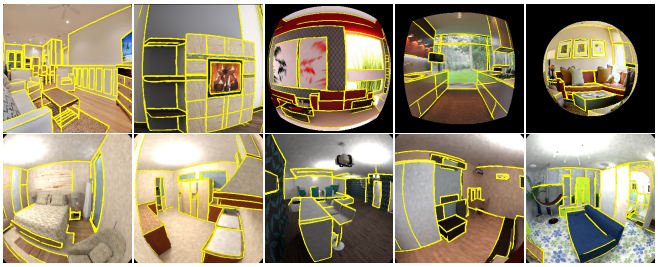


Fig. 4. Detection results of Distorted lines on the *D-Wireframe* (the first row) and *Fish-SUNCG* (the second row) datasets.

work. Then fed I_f together with $\hat{L_f}$ into the following rectification network to predict global and local distortion parameters, and a rough correction output I_r will be generated by the predicted distortion parameters. Considering that the uncertain errors in distorted line map $\hat{L_f}$ could result in the deviations of distortion parameters and I_r, an *Attention-based Refinement (AR)* module also designed for further compensating the rectification error and achieving a more robust optimization result.

3.1 Line-aware Rectification

LR module aims at learning the optimal distortion parameters under the guidance of structure lines. It mainly consists of three sub-modules: the Distorted Lines Perception (DLP) sub-module which solves the problem of distorted lines extraction; the *Multi-Scale Calibration sub-module which predicts distortion parameters by fusing both global and local feature; and the* Rectification sub-module which leverages predicted distortion parameters to rectify the input fisheye image I_f, and the details as follow.

Distorted Lines Perception Module. This module aims at extracting the geometric lines from the input RGB fisheye image, which could provide high dimensional information and implicit constraints for the following rectification step. Inspired by the traditional calibration method [12] which follows the rule of *the projection of a straight line from space to the camera should be a line* and the edge detection network [20], [35], we propose to encode this distorted lines with CNNs and also take this geometric rule as a self-supervised constraint in distortion correction. For a fisheye image $I_f \in \mathbb{R}^{H\times W\times 3}$ as input, we combine the Pyramid Residual Module (PRM) [36] and the Stacked Hourglass Module [37] to learn the *distorted lines* map $\hat{L_f} \in \mathbb{R}^{H\times W}$, where the H and W are width and height of the input image. In detail, we firstly use cascaded PRMs to extract feature maps of $\frac{H}{4} \times \frac{W}{4} \times 256$ from I_f, and then we use five stacked hourglass modules to obtain the high dimensional structure features, meanwhile keeping the feature map size unchanged. Finally, several deconvolution and convolution layers are adopted to predict the distorted line map. Note that the Batch-Normalization and the ReLU layers are used for each (de)convolution layer except for the last prediction layer. Pixel-wise definition of the target probability map $\hat{L_f}$ is shown as follow:

$$\hat{L_f}(\boldsymbol{p}) = \begin{cases} d(\boldsymbol{l}) & \text{if } \boldsymbol{p} \text{ is (nearly) on } \boldsymbol{l} \in \boldsymbol{L}, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\boldsymbol{L}$ denotes the set of *distorted lines*: $\boldsymbol{L} = \{\boldsymbol{l_1}, \ldots, \boldsymbol{l_n}\}$, and the value $d(\boldsymbol{l}_i)$ is defined as the length of line l if pixel $\boldsymbol{p}$ is located (nearly) on a distorted line segment; otherwise 0. $\hat{L_f}$ not only can indicate if pixel $\boldsymbol{p}$ is passed through a line, but also the predicted length implicitly contains the information for the distortion parameters. Several results of *DLP are shown in Fig. 4.*

Next, we concatenate the final output $\hat{L_f}$ from DLP and the original input I_f together as a whole, and then fed them into a convolution and the L1-L4 layers of ResNet-50 [38], as depicted in Fig. 2. The output feature map is served as the input of the following multi-scale calibration module, and the detailed description about this calibration module is in the next subsection.

Multi-Scale Calibration Module. Considering the fisheye distortion with the nonlinear distribution, the central area generally has less distortion while the degree of distortion in other regions is proportional to the distance away from the center. To solve this problem, a multi-scale calibration module is designed to compensate for the overall rectification effect of the whole fisheye image, as shown in Fig. 5.

For the input feature volume $\mathcal{F}_c \in \mathbb{R}^{h\times w\times c}$, where h, w, and c indicate height, width and channels respectively, it is the output from a convolution and the L1-L4 layers of ResNet-50, as shown in Fig. 2. And then a bifurcated structure with global and local branches is proposed in special to fit this nonlinearity in fisheye images. For the global branch, it treats the feature volume $\mathcal{F}_c$ as a whole to learn global features by a series of convolutional layers, along with two fully connected (FC) layers in succession. And the last FC layer output a 9-D vector which represents the global distortion parameters denoted by K_g. For the local branch, we break the feature volume $\mathcal{F}_c$ into a series of local blocks to estimate local distortion parameters respectively, and each block represents different local distortion information. Specifically, the $\mathcal{F}_c$ is divided into five blocks: the central region with the size of $6 \times 6 \times 1024$ and four marginal regions of upper left, lower left, upper right and lower right respectively with the size of $5 \times 5 \times 1024$. Then these five set of sub-features are sent into two FC layers and a linear filter separately to predict the local parameters, which denoted by $K_{loc} = \{K_{loc}^k\}_{k=1}^5$. The parameter settings of these two FC layers and the pooling layer are the same as those in the global branch, and meanwhile, these five local streams share the same weights. Since the parameters m_u, m_v and u_0, v_0 are related to the entire image, the linear filter only predicts the first five distortion parameters $k_1, \ldots, k_5$, which means each output K_{loc}^k is a 5-D vector.

In the training process, we define the final distortion parameters K_d as the average value of K_g and K_{loc}^k, which consider all the regions in the fisheye image and also balance the nonlinearity.

Rectification Module. This module serves as a bridge between the distortion parameters and geometry structures while transforming errors from the distortion parameter domain to the image domain. With the final distortion parameters K_d as the input, the rectification module can not only compute the corresponding undistorted pixel coordinates, but also generate the rectified fisheye image I_r with the rectified *distorted lines* mask I_{mask}. To be attention that

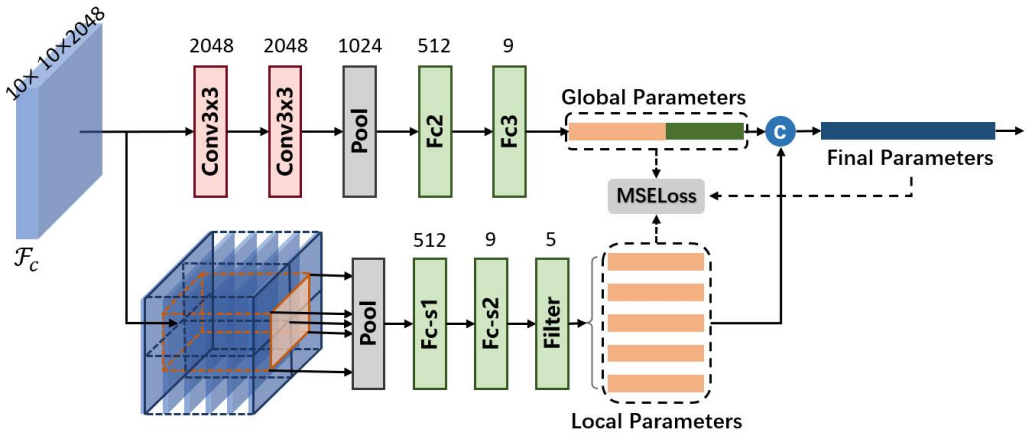


Fig. 5. Illustration of the proposed multi-scale calibration module. The final distortion parameters can be regressed through the global and local paths, which use the output feature map $\mathcal{F}_s$ of the previous *Resnet module as the input.*

the I_{mask} is indicated with the yellow highlights in I_r, as shown in Fig. 2. And the details of module implementation are elaborated as below.

In the forward propagation, supposing that the pixel coordinates in the original fisheye image I_f and the rectified image I_r are $\boldsymbol{p}_f = (x_f, y_f)$ and $\boldsymbol{p}_r = (x_r, y_r)$ respectively. Then the relationship between them can be expressed as:

$$\boldsymbol{p}_f = \mathcal{T}(\boldsymbol{p}_r, K_d) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \frac{r(\theta)\boldsymbol{p}_d}{\|\boldsymbol{p}_d\|_2} \tag{4}$$

where $\mathcal{T}$ indicates the forward fisheye projection function. During the rectification process, if the point $\boldsymbol{p}_f$ belongs to distorted lines $\boldsymbol{L}$ then adding it into the mask I_{mask}.

Since the rectification process involves scaling, bilinear interpolation is adopted to avoid pixel missing in the rectified image. Denoting the process as $\mathcal{R}$, we can get:

$$\begin{aligned} \mathcal{R}(\boldsymbol{p}_f) =& \overline{w_x}\,\overline{w_y} I_f(\lfloor x_f \rfloor, \lfloor y_f \rfloor) + \overline{w_x} w_y I_f(\lfloor x_f \rfloor, \lceil y_f \rceil) \\ &+ w_x \overline{w_y} I_f(\lceil x_f \rceil, \lfloor y_f \rfloor) + w_x w_y I_f(\lceil x_f \rceil, \lceil y_f \rceil), \end{aligned} \tag{5}$$

where w_x and w_y describe the distance-weighted coefficients: $w_x = x_f - \lfloor x_f \rfloor$, $w_y = y_f - \lfloor y_f \rfloor$; $\overline{w_x} = 1 - w_x$, and $\overline{w_y} = 1 - w_y$. For a coordinate point $\boldsymbol{p}_r(x_r, y_r)$ in the corrected image I_r, its corresponding pixel value is calculated as $I_r(x_r, y_r) = \mathcal{R}(p_f)$.

To verify the feasibility of this module, we also prove that the rectification module is differentiable as well as the derivative of rectified image I_r with respect to the distortion parameters K_d is propagable, as follows:

$$\begin{aligned} \frac{\partial I_r}{\partial K_d^i} &= \frac{\partial I_r}{\partial x_f} \cdot \frac{\partial x_f}{\partial K_d^i} + \frac{\partial I_r}{\partial y_f} \cdot \frac{\partial y_f}{\partial K_d^i} \\ &= \mathcal{R}'_{x_f} \mathcal{T}'_{K_d^i} + \mathcal{R}'_{y_f} \mathcal{T}'_{K_d^i}, \end{aligned} \tag{6}$$

$\mathcal{R}'_{x_f}$ is the partial derivative of interpolation function $\mathcal{R}$ in the x direction, and $\mathcal{R}'_{y_f}$ is in the y direction. $\mathcal{T}'_{K_d^i}$ is the partial derivative of projection function $\mathcal{T}$ with respect to each distortion parameter.

3.2 Attention-based Refinement

Given that the distorted lines $\hat{L_f}$ extracted from the *DLP module are not accurate for all the pixels and the wrong extract results may likely lead to wrong guidance during the following training phase, the* AR module is designed specially to learn the probability that the distorted lines being detected incorrectly, as shown in Fig. 2. This uncertain probability is proposed by [39] that uses it in multi-task learning, and here we modify and introduce it into fisheye image rectification task for solving the noisy of *distorted lines*. We use this uncertain map to constrain the error between each pixel and the truth value, meanwhile get better optimization effect.

We first select the intermediate features from each stage's output of Resnet (containing four stages: L1-L4), and then put each of them through a convolution layer with the 1×1 kernel size for downscaling the number of channels. After that, bilinear up-sampling operations with corresponding scale factors are used to obtain four groups of feature maps in the fixed size: $F_{C_i} \in \mathbb{R}^{\mathbb{H}\times\mathbb{W}\times\mathbb{C}}(i = 1, 2, 3, 4)$. Then the input RGB fisheye image I_f, the generated rectified image I_r, as well as all the multi-channel features F_{C_i} are concatenated as a new feature map denoted as F'_C, which is fed into a convolution layer followed by a softmax activation function to predict the uncertain map U_A:

$$U_A = Softmax(F'_C W_A + b_A), \tag{7}$$

where the $Softmax(\cdot)$ is a channel-wise softmax function used for the normalization.

As shown in Fig. 6, the uncertainty of all pixels in generated perspective image is uniform at the beginning of training. With the increase of the number of the epoch, the uncertainty of U_A decreases obviously that the uncertain pixel loss is more concentrated on the areas with rich textures or large gradients, which means the *AR module learns the probability of error in pixels and the* LR module does have the ability to rectify distortion.

3.3 Overall Optimization Objective

For any input image I_f, proposed FirNet can output the distorted line map $\hat{L_f}$, distortion parameters K_g, $K_{loc}^k, k = 1, \ldots, 5$ and K_d , as well as the rectified fisheye image I_r with distorted lines mask I_{mask} in an end-to-end manner. During the training phase, we employ different losses as supervision signals for different modules.

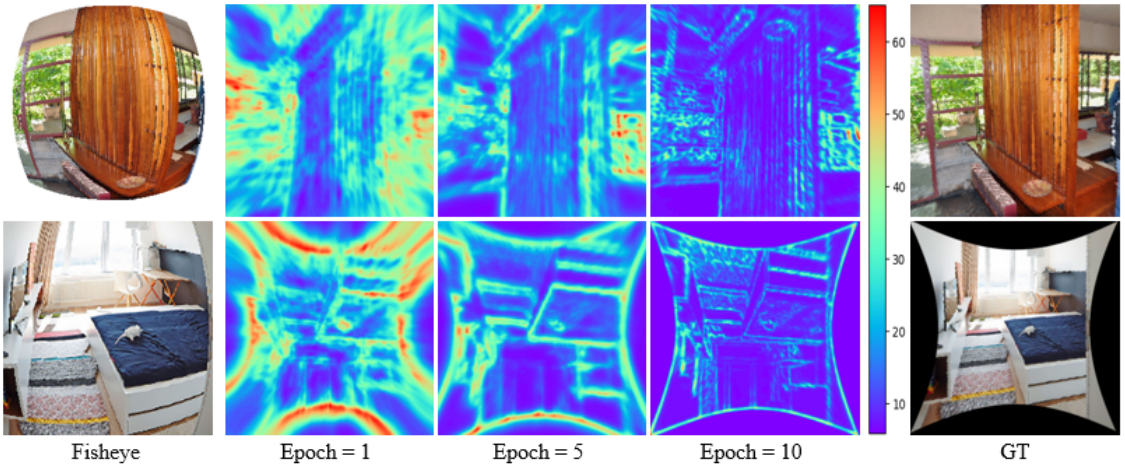


Fig. 6. The uncertain map U_A learned from the attention-based refinement module. **Left:** RGB fisheye images. **Middle:** the learned uncertain map U_A in different epochs. **Right:** The ground truth images.

Loss of Distorted Line Map Learning. Considering the fact that distorted line segments are 0-measure geometric primitives in 2D images, most of pixels for the target $\hat{L_f}$ defined in Eq. (3) will be 0. In other words, most of pixels will not be passed through any distorted line segment. For the sake of representation simplicity, the pixels not on any distorted line segment are treated as the negative class Ω^- and the rest of pixels are collected in the positive class Ω^+, where $\Omega^+ = \Omega - \Omega^-$. Then, we weight these two classes in the loss function as

$$\mathcal{L}_{line} = \frac{|\Omega^-|}{|\Omega|} \sum_{\boldsymbol{p} \in \Omega^+} \mathcal{D}(\boldsymbol{p}) + \frac{|\Omega^+|}{|\Omega|} \sum_{\boldsymbol{p} \in \Omega^-} \mathcal{D}(\boldsymbol{p}), \tag{8}$$

where $\mathcal{D}(\boldsymbol{p}) = \|L_f(\boldsymbol{p}) - \hat{L_f}(\boldsymbol{p})\|_2^2$. The L_f is the ground truth.

Loss of Distortion Parameter Estimation. In the multi-scale calibration module, the global parameters K_g, local parameters K_{loc}, and the fused distorted parameters K_d are learned from the bifurcate structure, and they can capture global and local features to balance the nonlinear distortion distribution of fisheye images. Theoretically, the closer the regression distortion parameters are near to the ground-truth, the better the image rectification quality will be. So we use MSE loss to calculate the distance to supervise the distortion parameter optimization in the right direction. For the output K_g, we define the global loss $\mathcal{L}_{glo}$ as below:

$$\mathcal{L}_{glo} = \frac{1}{9} \sum_{i=1}^{9} w_i (K_g(i) - K_{gt}(i))^2, \tag{9}$$

where the $K_{glo}(i)$ and $K_{gt}(i)$ are the i-th component of predicted global parameters K_{glo} and the ground truth parameters K_{gt}. The weight w_i is used to rescale the magnitude between different components of distortion parameters. On the other side of the bifurcate, the loss of estimated parameters are defined as:

$$\mathcal{L}_{loc}^k = \frac{1}{5} \sum_{i=1}^{5} w_i (K_{loc}^k(i) - K_{gt}^k(i))^2, \tag{10}$$

where the $K_{loc}^k(i)$ is the i-th component of predicted local parameters K_{loc}^k. Similarly, the loss $\mathcal{L}_f$ of fused distortion parameters K_d could be also calculated, and the overall loss of parameters $\mathcal{L}_{para}$ can be denoted as:

$$\mathcal{L}_{para} = \lambda_f \mathcal{L}_f + \lambda_{glo} \mathcal{L}_{glo} + \lambda_{loc} \sum_{k=1}^{5} \mathcal{L}_{loc}^k, \tag{11}$$

where the λ_f, λ_{glo} and λ_{loc} are weight parameters to balance the different losses.

Loss of Geometric Constraints. Although the $\mathcal{L}_{para}$ enforces the network to fit the optimal distortion parameters, it is prone to get stuck in the local minimums when only using $\mathcal{L}_{para}$. Therefore, considering that the geometric structure could provide a stronger constraint to boost performance, we design a function $\mathcal{S}(\cdot,\cdot,\cdot)$ to calculate the geometric errors between the rectified image I_r (using fused distortion parameters K_d) and the ground truth image I_{gt} (using ground truth parameters K_{gt}):

$$\mathcal{S}(\boldsymbol{p}_f, K_d, K_{gt}) = Abs(\mathcal{F}(\boldsymbol{p}_f, K_d) - \mathcal{F}(\boldsymbol{p}_f, K_{gt})), \tag{12}$$

where $Abs(\cdot)$ is the absolute value operation, $\boldsymbol{p}_f$ is the pixel coordinates of fisheye image I_f, and $\mathcal{F}$ is the inverse function of $\mathcal{T}$ described in Eq. (4).

Besides, the mask I_{mask} contains the distorted lines that could provide geometric information in high dimensional of fisheye image, and also the non-zero pixels in I_{mask} are meaningful which invisibility indicate the degree of geometric structure distortion. The geometric error of the pixels in I_{mask} could accelerate and optimize network training, therefore, we pay more attention to the distortion in mask I_{mask} when calculating the geometric loss of the whole image. For a pixel $\boldsymbol{p}_d$ in the I_{mask}, we weight its geometric loss $\mathcal{L}_g$ with the coefficient λ_{mask} to guide the better structural optimization.

$$\mathcal{L}_g = \frac{1}{N} \Big(\sum_{\boldsymbol{p}_f \in \Omega^+} \lambda_{mask} \mathcal{S}(\boldsymbol{p}_f, K_d, K_{gt}) + \sum_{\boldsymbol{p}_f \in \Omega^-} \mathcal{S}(\boldsymbol{p}_f, K_d, K_{gt}) \Big), \tag{13}$$

where $\mathcal{F}$ is the inverse function of $\mathcal{T}$ which described in Eq. (4), and N is the number of all pixels that belong to I_r.

In general, the geometric loss $\mathcal{L}_g$ contains two main parts: the former is the loss of I_{mask} which controls the overall effect, while the latter focuses on none-mask pixels which contains more rectification details.

Loss of Uncertain Pixel. For the quality of image generation, we calculate the similarity between the rectification image I_r and the ground truth image I_{gt} at the pixel level, denoted as $\mathcal{L}_{pix} = \|(I_r - I_{gt})\|_1$. However the detected $\hat{L_f}$ would introduce the errors into the rectification image I_r, which could guide to a wrong direction when optimizing $\mathcal{L}_{pix}$, so we use the uncertain map U_A obtained from the attention-based refinement module to eliminate these uncertain errors and achieve more robust optimization:

$$\mathcal{L}_{pix} \leftarrow \mathcal{L}_{pix}/U_A + \log U_A, \tag{14}$$

Overall Loss. Our proposed FirNet consists of two phases during the training. In the first phase, we optimize the distorted lines perception from scratch with the loss function defined in Eq. (8), and we fix the parameters of *DLP* module once well trained. Then the remaining modules are trained to learn the distortion parameters in the second phase. The total losses we use here are defined as

$$\mathcal{L} = \lambda_{para}\mathcal{L}_{para} + \lambda_g\mathcal{L}_g + \lambda_{pix}\mathcal{L}_{pix}, \tag{15}$$

where λ_{para}, λ_g, and λ_{pix} are weight parameters to balance the different terms. The detailed steps of the optimization are shown in the Section. 5.1

4 Dataset for Calibration

There still remains a crucial problem for training the proposed neural network that the real distortion parameters as well as well-annotated distorted and rectified line maps are required. However, to the best of our knowledge, there is no such a large scale dataset that can satisfy all the above requirements. Thanks to the recently released wireframe dataset [20] which has the labelings of straight line and the large-scale 3D scenes SUNCG [21] which provides diverse semantic 3D scenes, we construct a new dataset with well-annotated 2D/3D line segments $\boldsymbol{L}$ as well as the corresponding distortion parameters K_{gt} for training and testing. The two subsets of our datasets, the distorted wireframe collection *(D-Wireframe)* from the wireframe dataset and the fisheye SUNCG collection (*Fish-SUNCG*) from the 3D model repository, are described in Fig. 7.

Distorted Wireframe Collection (D-Wireframe). For any image in the wireframe dataset proposed by [3] which contains 5462 normal perspective images marked with straight line segments, we randomly generate eight different sets of distortion parameters to transform this perspective image into a fisheye image with different distortion effects by Eq. (1). Thus, the perspective image and the corresponding line segment annotations can be distorted to the fisheye image with distorted line segments. In summary, we generate this collection $\mathcal{D}_{wf}$ and split it into the training set and the testing set with $40,000$ and 1848 samples respectively.

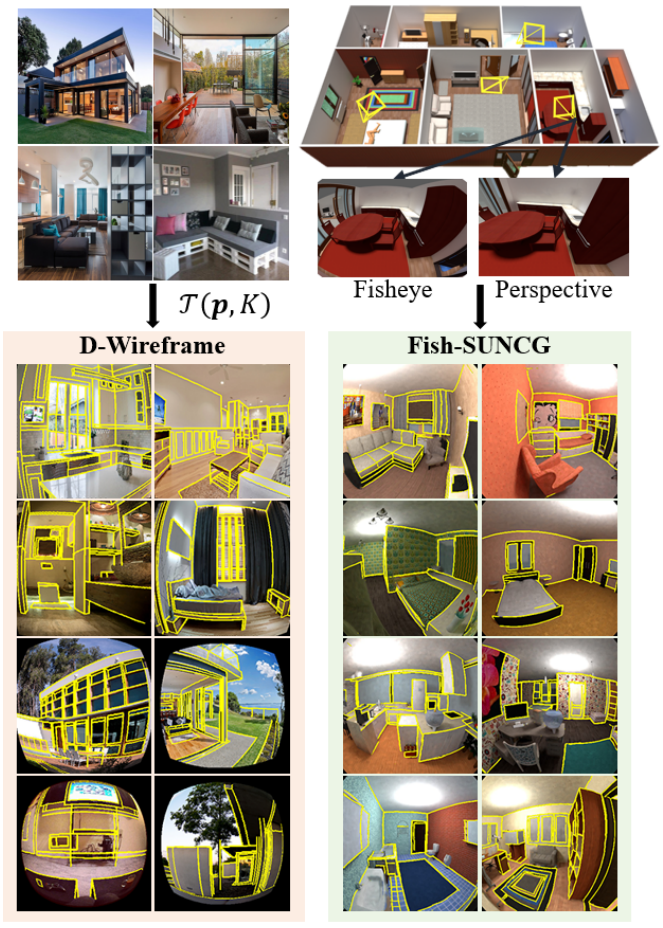


Fig. 7. Data samples from the distorted wireframe (left) and the fisheye SUNCG collection (right). For D-Wireframe, the normal images can be converted to any type of fisheye images using random distortion parameters K; for Fish-SUNCG, given a randomly selected perspective from the virtual rendering scene, the fisheye projection image and the perspective image under this view can be generated simultaneously.

Fisheye SUNCG Collection (Fish-SUNCG). The D-wireframe collection can provide sufficient fisheye distortion images in rich diversity and flexibility for the network training. However, artificially distorting the images that are taken by perspective cameras cannot fully characterize the fisheye distortion for real scenarios. We address this problem by simulating the various image formation in both perspective and fisheye cameras at the same observation positions from the 3D models of SUNCG [21] which contains 45K different virtual 3D scenes with manually created realistic room and furniture layouts. In detail, we use the Blender [40] to render images by specifying the camera pose and imaging formation model. The rendering protocol is illustrated in Fig. 7. For the line segment generating, we remove the texture of 3D models to get the wireframe model of 3D objects. After that, we manually remove the edges of wireframe manually to get the line segments that are on the boundary of objects. Since we are able to control the image formation without metric errors, the data samples can be used to train our network without information losses. In the end, we generate 6,000 image pairs from 1,000 scenes for training and 300 image pairs from 125 scenes for testing. This collection is denoted as $\mathcal{D}_{sun}$.

5 Experiments

5.1 Implementation Details

We use the distorted fisheye image and the corresponding line segment map pairs of distorted wireframe collection $\mathcal{D}_{wf}$ to train the *DLP module in the first step. After that, we fix the weights of* DLP module and train the rest of the rectification network by using our proposed $\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$ together. The input size for our network is set as 320×320 for both training and testing phases.

The weight parameters in Eq. (15) are set to as follows: $\lambda_g = 100$, $\lambda_{pix} = \lambda_{para} = 1$, and the balance parameters are set to as follow: $W = \{w_1 = 0.1, w_2 = 0.1, w_3 = 0.5, w_4 = 1, w_5 = 1, w_6 = 0.1, w_7 = 0.1, w_8 = 0.1, w_9 = 0.1\}$. The optimization method we use for the training is the stochastic gradient descent method (SGD). The initial learning rate is set to 0.0001, and then it is decreased by a multiple of 0.1 after 20 epochs. And our network is implemented on the PyTorch platform with a single Titan-X GPU device.

5.2 Evaluation Metrics

Benefitted from the *DLP* module of our proposed model, we are able to evaluate the performance on geometry of the rectified lines that have eliminated the distortion by comparing the rectified distorted line map $\hat{L_r}$ from the rectification module and the ground truth line map L_{gt} with real distortion parameters. What's more, the *Precision* and *Recall* are redefined to quantitatively evaluate the error between $\hat{L_r}$ and L_{gt}. Further more, the reprojection error (RPE) is proposed to evaluate the overall rectification effects by measuring the deviation of the pixel between the rectified image and the fisheye image. On the other hand, we also follow the evaluation metrics used in previous works [18], [19] that utilize the peak signal to noise ratio (PSNR) and structure similarity index (SSIM) for evaluating the rectified images.

Precision v.s. Recall. The precision and the recall rate of the predicted line segment map is defined as

$$Precision = |P \cap G|/|P|, \quad Recall = |P \cap G|/|G|, \tag{16}$$

where the P is the set of edge pixels in the rectified line segment map $\hat{L_r}$ and G is the set of edge pixels in the ground truth of line segment map L_{gt} without distortion. We evaluate the performance of algorithms with F-value: $F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

PSNR and SSIM. These two metrics are widely used to describe the degree of pixel blurring and structure distortion respectively, so we use them here to compare the rectified fisheye images. In general, the larger values of PSNR and SSIM, the better the rectification quality.

Reprejection Error (RPE). This metric is generally used to quantify the distance between an estimation of a 2D/3D point and its true projection point. So the real distortion parameters K_{gt} and the estimated ones K_d are used to rectify the pixels $\boldsymbol{p}_f$ of the fisheye image, and we can get the projection images $\mathcal{F}(\boldsymbol{p}_f, K_{gt})$ and $\mathcal{F}(\boldsymbol{p}_f, K_d)$ respectively, where the $\mathcal{F}$ is the function representation of Eq. (12). The mean square error (MSE) of the RPE in the whole fisheye image is defined by $\frac{1}{N}\sum_{\boldsymbol{p}_d \in \Omega}(\mathcal{F}(\boldsymbol{p}_f, K_{gt}) - \mathcal{F}(\boldsymbol{p}_f, k_d))^2$.

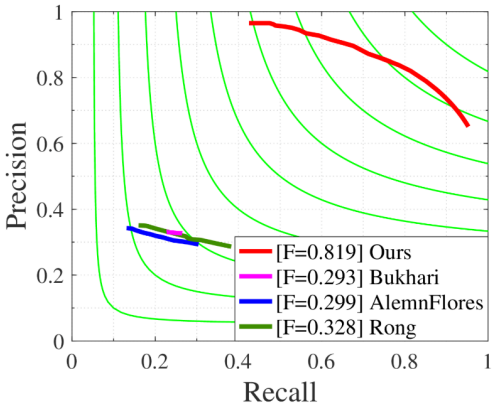


Fig. 8. The precision-recall curves of different rectification methods for the rectification of distortion line maps [15], [16], [18].

5.3 Compared with State-of-the-art

Devoted to further excavating and using more effective geometric information, we propose a novel fisheye rectification network with the self-attention mechanism, which uses distorted lines as geometric constraints to rectify the fisheye distortion. Given an arbitrary single fisheye image, the rectified image and corresponding distortion parameters could be obtained through our proposed network. And then the quantitatively and qualitatively evaluations are performed in our proposed synthesized datasets ($\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$), as well as on the real world fisheye image to further demonstrate the feasibility and validity of our proposed method. And more experimental results could be seen in http://captain.whu.edu.cn/FirNet

Results on Geometry Rectification. Our proposed model always follows the geometric projection principle: *the projection of straight line from space to the camera should be a line* that it tries to learn geometric constraints to guarantee the rectification effects. In other words, if the rectified distorted line map still exists curved geometries or has deviations from the ground truth, it means that the learned distortion parameters are not accurate enough and the overall structure rectification is unsatisfactory. Visually, we show the rectified distorted line maps which are output from the rectification module in Fig. 9 to verify that our network indeed has the ability to recover the straight line characteristics. The results prove the validity of the geometric constraint in our network in which the rectified line map through our network is indeed straightened, while those rectified by other methods are still distorted to some extent.

Furthermore, additional qualitative experiments are performed to compare our method with [15], [16], [18], as shown in Fig. 8. The precision-recall curves show that the rectified distortion lines obtained by our proposed network is the closest to its original geometry and far more than other methods in the accuracy of F-value (F-value =.819), which also demonstrates that our proposed FirNet is far superior to other methods in restoring the geometric structure of images.

Results on Synthetic Dataset. The synthetic datasets $\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$ created by us provide a rich test sample with

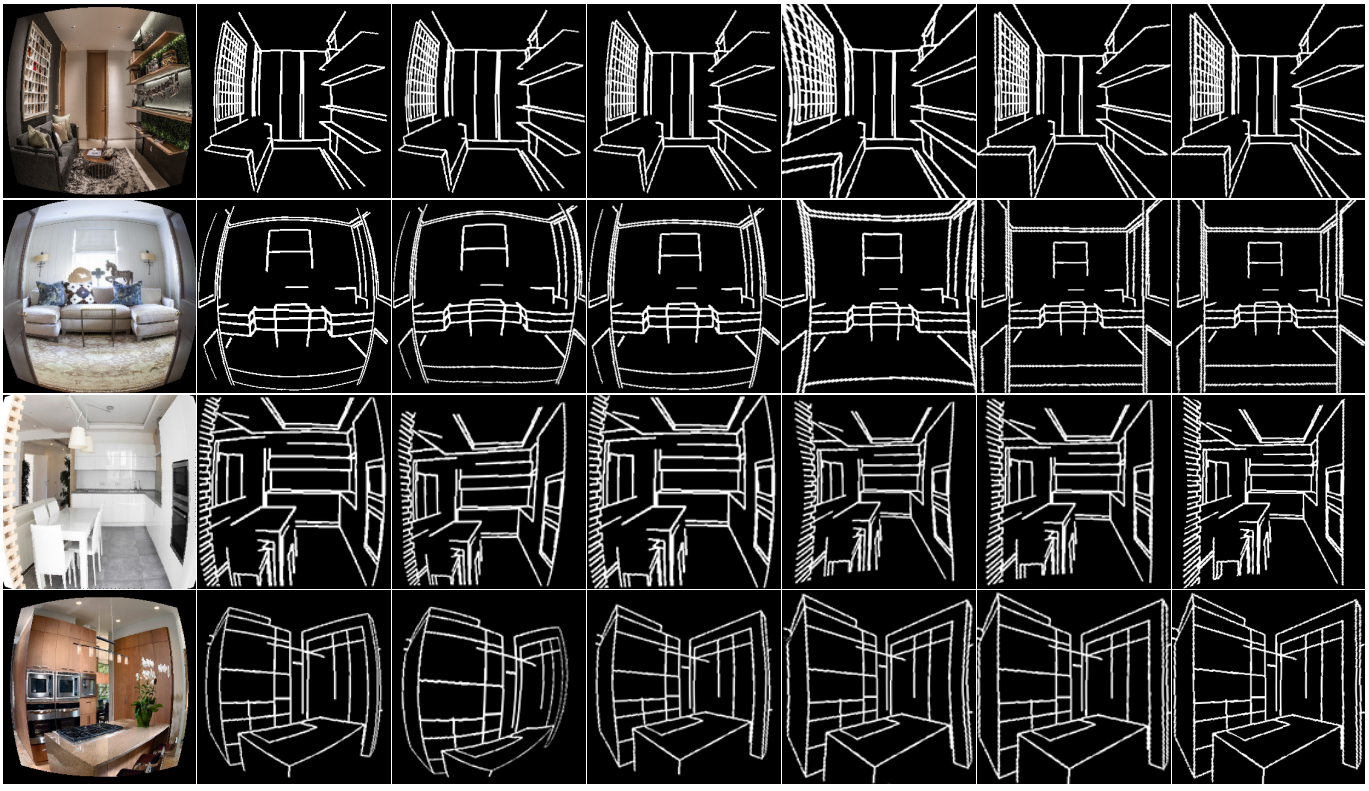


Fig. 9. Distortion line rectification results of various methods. From left to right are the input RGB fisheye images, the distorted lines detected in fisheye images, the rectified results by different method [15], [16], [18], our proposed method, and the ground truths.

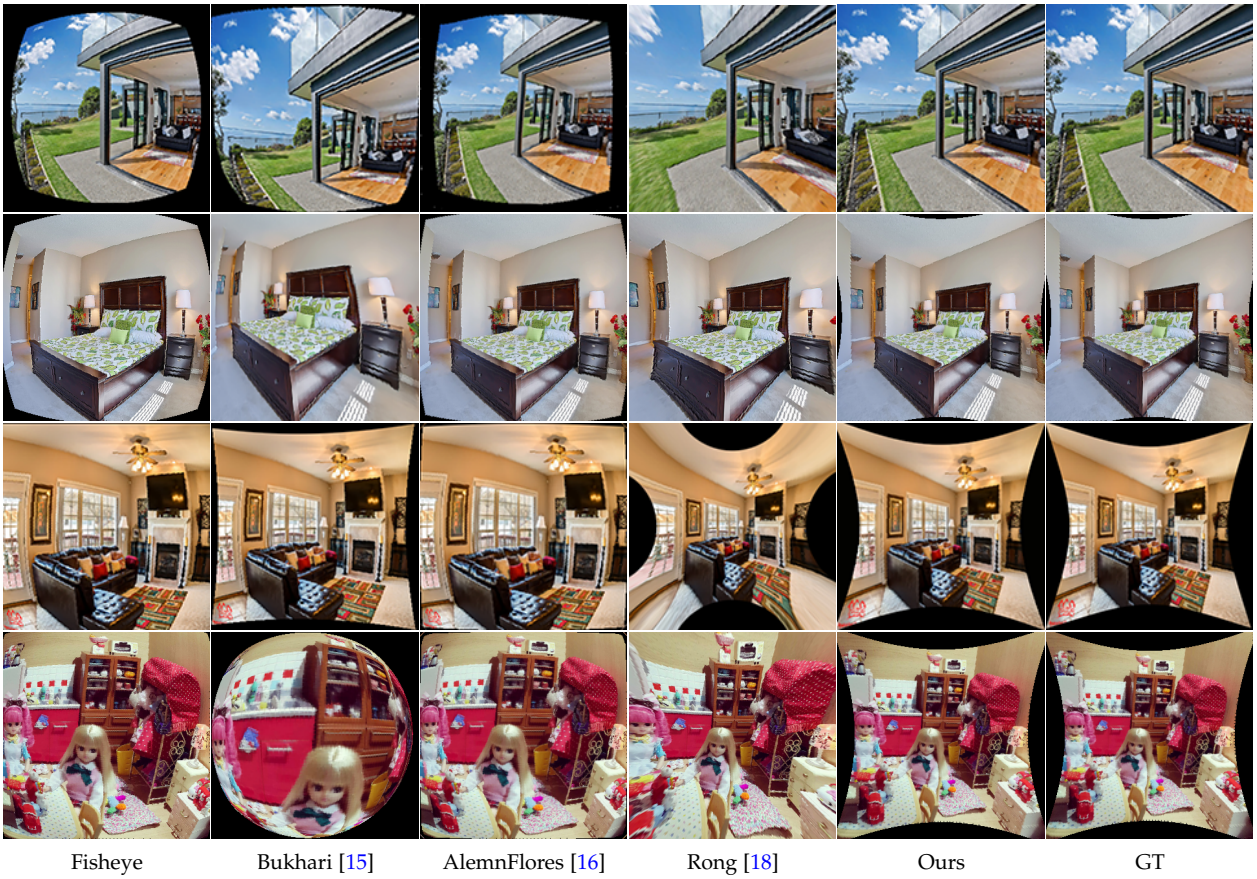
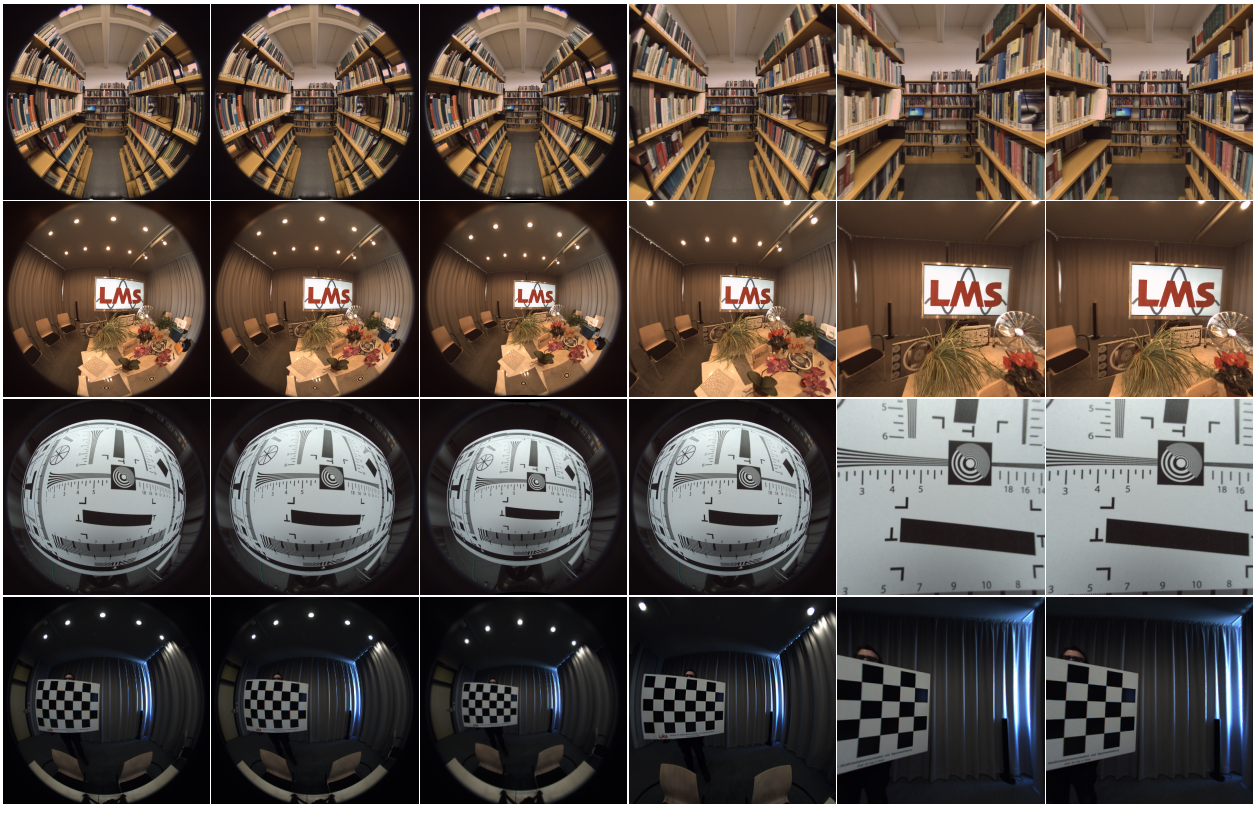


Fig. 10. Qualitative comparison results of fisheye image rectification on $\mathcal{D}_{wf}$. From left to right are the input fisheye images, rectification results of three state-of-the-art methods (Bukhari [15], AlemnFlores [16], Rong [18]), our results as well as the ground truth images.

Fisheye Bukhari [15] AlemnFlores [16] Rong [18] Ours GT

Fig. 11. Qualitative comparison results of fisheye images rectification on $\mathcal{D}_{sun}$. From left to right are the input fisheye images, rectification results of three state-of-the-art methods (Bukhari [15], AlemnFlores [16], Rong [18]), our results as well as the ground truth images.



Fisheye Bukhari [15] AlemnFlores [16] Rong [18] Ours GT

Fig. 12. Qualitative rectification comparison results on real fisheye dataset [41]. From left to right are the input fisheye images, rectification results of three state-of-the-art methods (Bukhari [15], AlemnFlores [16], Rong [18]), our results as well as the ground truth images.

TABLE 1
Comparison results with the state-of-the-arts on $\mathcal{D}_{wf}$, $\mathcal{D}_{sun}$, and the fused dataset ($\mathcal{D}_{wf} + \mathcal{D}_{sun}$) respectively, using the PSNR, SSIM, and reprojection error (RPE) metrics. (*) These results are reported in [19].

Method	D_{wf}			D_{sun}			$D_{wf} + D_{sun}$		
	PSNR	SSIM	RPE	PSNR	SSIM	RPE	PSNR	SSIM	RPE
Bukhari [15]	10.84	0.19	161.33	9.16	0.14	167.12	9.34	0.18	164.75
AlemnFlores [16]	10.73	0.26	126.29	9.98	0.22	124.41	10.23	0.26	125.43
Rong [18]	12.78	0.31	122.54	13.96	0.33	120.16	12.92	0.32	121.69
FishRectNet [19]	14.96*	0.41*	-	-	-	-	-	-	-
Ours	**25.19**	**0.79**	**0.45**	**46.32**	**0.99**	**0.21**	**28.06**	**0.90**	**0.33**

TABLE 2
Training on $\mathcal{D}_{wf}$, $\mathcal{D}_{sun}$, and the fused dataset ($\mathcal{D}_{wf} + \mathcal{D}_{sun}$) respectively, while testing on the real fisheye dataset [41] with the metrics of the PSNR, SSIM and reprojection error (RPE). For the traditional calibartion metheds [15], [16], the evaluation results are independent of $\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$.

Method	D_{wf}			D_{sun}			$D_{wf} + D_{sun}$		
	PSNR	SSIM	RPE	PSNR	SSIM	RPE	PSNR	SSIM	RPE
Bukhari [15]	9.84	0.16	156.32	9.84	0.16	156.32	9.84	0.16	156.3
AlemnFlores [16]	10.72	0.30	129.15	10.72	0.30	129.15	10.72	0.30	129.1
Rong [18]	11.75	0.30	125.73	10.49	0.20	160.5	11.81	0.30	125.31
FishRectNet [19]	-	-	-	-	-	-	-	-	-
Ours	**19.86**	**0.67**	**1.91**	**11.34**	**0.39**	**122.12**	**22.34**	**0.82**	**1.68**

different distortion types. We follow the evaluation metrics PSNR, SSIM as well as RPE to quantitatively evaluate rectified fisheye images on $\mathcal{D}_{wf}$, $\mathcal{D}_{sun}$, and fused dataset ($\mathcal{D}_{wf}$ + $\mathcal{D}_{sun}$) respectively, as reported in Tab. 1. From the evaluation results, it demonstrates that no matter in image rectification or structure maintenance, our method is obviously superior to other methods and has achieved the highest score on PSNR, SSIM as well as RPE. It is worth mentioning that the reprojection error of the whole image calculated by our method is within one pixel, while other methods far behind us.

For the visual rectification effects, we visualize the rectified images by our method and start-of-the-art methods [15], [16], [18] on the test set of $\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$ respectively, as shown in Fig. 10 and Fig. 11. For $\mathcal{D}_{wf}$, we selected special images with different types of fisheye distortion, such as typical full-frame fisheye images, full circle fisheye images, and drum fisheye images. The results show that our method has an excellent rectification effect in visual effects, while other methods can not satisfy the needs of correcting distortion images with various types of fisheye distortion. For $\mathcal{D}_{sun}$, our network also achieves the best rectification performances and the rectified fisheye images are closer to ground truth than others.

Results on Wild Fisheye Images. To verify the validity of the proposed method, we also perform rectification experiments on real-world fisheye images compared with the traditional calibration methods [15], [16] and the CNN-based method [18]. As shown in Fig. 12, fisheye sequences of different real world scenes as well as calibration images with checkerboards are provided [41]. To calibrate the sequence of calibration fisheye images with checkerboard patternes, we use the calibration toolbox [42] based on the 2D calibration pattern to estimate internal and external parameters, and treat the rectification results obtained from [42] as ground truths. It can be concluded that our proposed method has the best performance in distortion rectification while other methods are hard to deal with this task, and a quantitative experiment in Tab. 2 also proves it. And because of the authenticity of $\mathcal{D}_{sun}$ and the diversity of $\mathcal{D}_{wf}$, the model trained on the fused dataset achieves the best rectification performance. Since traditional calibration methods [15], [16] are only related to the fisheye images to be tested, the same evaluation results are placed for different dataset comparisons.

Besides, since images in [41] are limited in distortion effect from the same fisheye camera, we also select various fisheye images with different distortion effects from the Internet to evaluate our model's performance. As shown in Fig. 13, it demonstrates that our proposed method has an excellent rectification performance even for real fisheye images with different distortion effects, which proves that our network has a higher rectification ability.

5.4 Ablation Study

TABLE 3
Ablations study of the proposed method with training and testing on the fused dataset ($\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$). The $BL_1 - BL_6$ represent different training strategies respectively.

Baseline	Setup	PSNR	SSIM	RPE
BL_1	$I_f \xrightarrow{base} I_r$	16.24	0.61	4.51
BL_2	$\hat{L_f} \xrightarrow{base} I_r$	18.61	0.62	2.08
BL_3	$[I_f, \hat{L_f}] \xrightarrow{base} I_r$	21.29	0.68	1.35
BL_4	BL_3 + Multi-Scale Calibration	25.65	0.80	0.91
BL_5	BL_4 + Geometric Constraints Loss	27.83	0.88	0.43
BL_6	BL_5 + Uncertain Pixel Loss	**28.46**	**0.90**	**0.33**

We conduct ablation studies on fused synthetic dataset, including $\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$. As shown in Tab. 3, six baselines (BL_1, BL_2, BL_3, BL_4, BL_5, BL_6) are designed to illuminate the rationality and effectiveness of each module in the proposed network. In detail, BL_1 only uses the ResNet L1-L4 to learn distorted parameters with RGB fisheye image

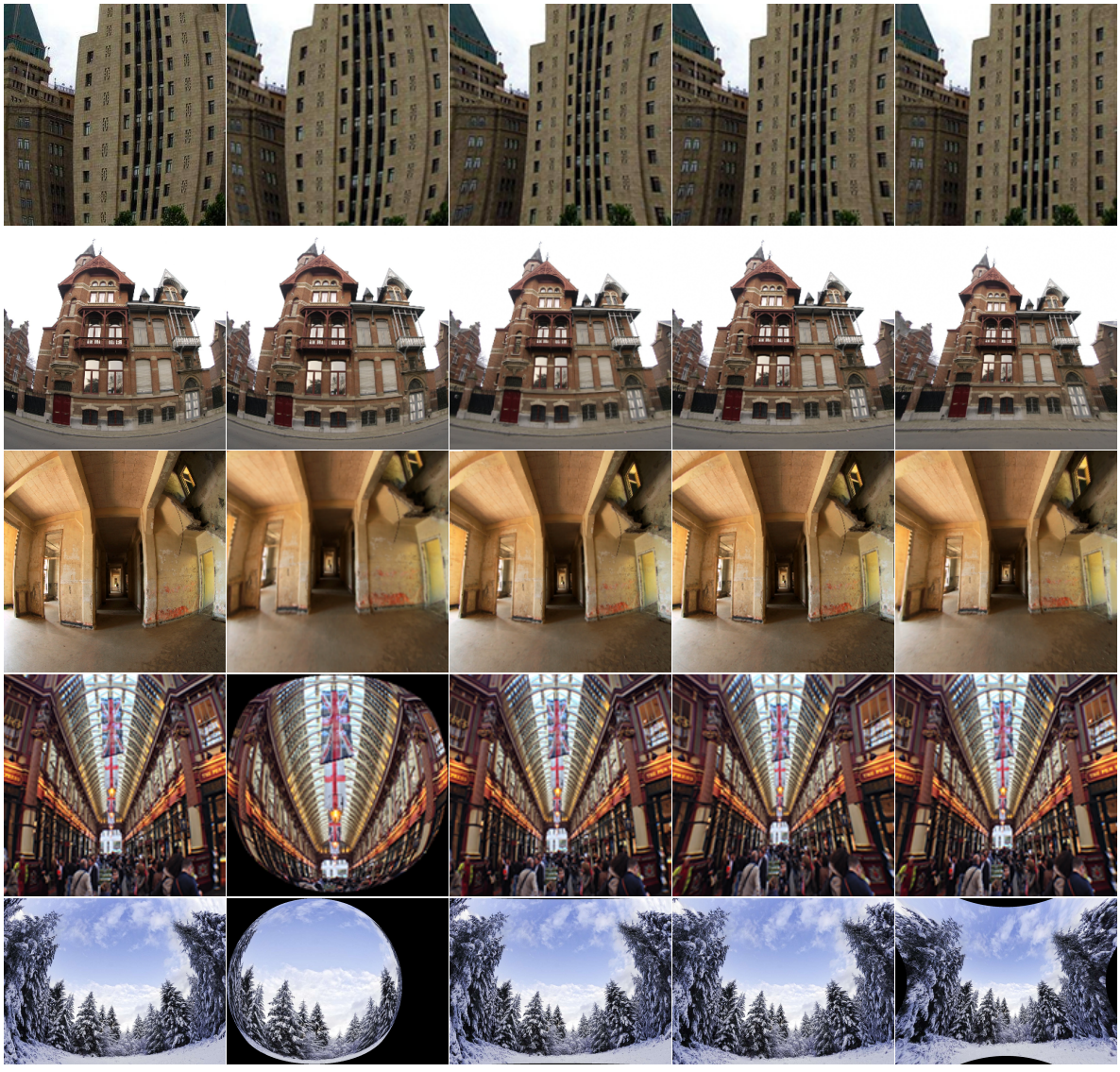


Fig. 13. Qualitative rectification comparison results on wild fisheye images with different distorted effects which mined from Internet. From left to right are the input fisheye images, rectification results of three state-of-the-art methods (Bukhari [15], AlemnFlores [16], Rong [18]), and our results.

input I_f; BL_2 uses the same network structure as BL_1 but with additional distorted lines $\hat{L_f}$ as the input; BL_3 also adopts the simplest networks with a combined input of I_f and $\hat{L_f}$; BL_4 adds the multi-scale calibration module based on BL_3 to compensate the non-linearity of the distortion distribution; BL_5 adds the geometric constraint loss $\mathcal{L}_g$ to guarantee the similarity of geometric structure between generated images and ground truth images; BL_6 uses the novel uncertain map $\mathcal{L}_{pix}$ to implement a more robust optimization. All the baselines are trained and tested on the same fused dataset ($\mathcal{D}_{wf}$ and $\mathcal{D}_{sun}$).

The results of ablation study are shown in Tab. 3. We can observe that the rectification performance of the BL_1 is disappointing because of the simple network structure, and the BL_2 is better than BL_1 since the distorted line map $\hat{L_f}$ learned from *DLP* introduces more efficient geometric structure information. What's more, compared with the BL_1 and BL_2, the BL_3 with the combined inputs (I_f and $\hat{L_f}$) further improves the performance with an additional guidance of geometric structures, which confirms the importance of distorted lines in fisheye rectification task. Due to the nonlinear distortion distribution of fisheye images, the BL_4 introduces more balanced distortion parameters P_d that are obtained from the multi-scale calibration module which takes into account the global and local distortion differences, and naturally the result shows that the balanced distortion parameters have a great improvement for overall rectification quality. And the BL_5 exploits the geometric constraint in distortion parameter learning with the supervision of the geometric loss and outperforms all the previous baselines, which demonstrates the feasibility and effectiveness of applying strong geometric constraints to rectification learning. The BL_6 performs best in rectification performance, which incorporates all the strategies, including multi-scale calibration, geometric constrains loss as well as uncertainty-guided pixel loss.

For qualitative evaluations, we also conduct comparison experiments as shown in Fig. 14, for intuitively observing

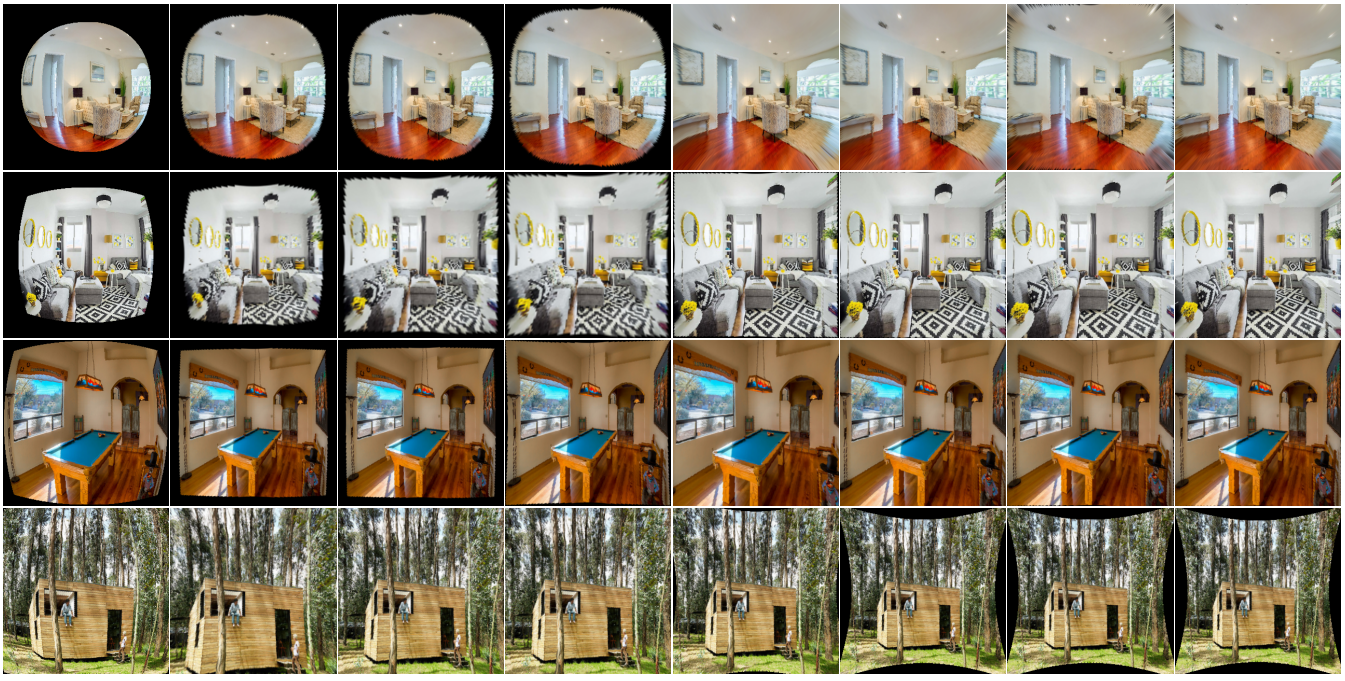


Fig. 14. Visulization of ablation experiments, and the $BL_1 - BL_6$ represent different baselines respectively like Tab. 3.

the rectification effects of different baselines. It is obvious that the rectification results are significantly improved with the distorted line $\hat{L_f}$ as the part of input, and the rectification effects of BL_3 and Ll_4 will be unstable once lacking of multi-scale calibration module or geometric constraint loss. Besides, the problems of over-rectified and under-rectified may occur if the aforementioned components are not used. Although there are extremely few differences between BL_5 and BL_6 visually, BL_6 can approach the ground truth quickly and achieve a more robust optimization.

6 Conclusion

In this paper, we propose a novel and efficient model name FirNet which utilizes line constraints to calibrate the fisheye lenses and eliminate the distortion parameters automatically only from a single image. Specifically, we adopt Line-aware strategies and multi-scale calibration module to learn how to straighten the distorted lines, and an attention-based refinement module is also proposed to obtain uncertain maps which serve as an uncertainty-guided pixel loss to solve the inaccurate distorted line labels issue. Extensive experimental results on synthesized datasets as well as real fisheye images demonstrate that our method performs much better results than present state-of-the-art. To better train the proposed network, we also reuse the existing datasets that have rich 2D and 3D geometric information to generate a synthetic dataset for fisheye calibration.

Since the geometric constraints of a single image are limited and the data distribution of the synthesized dataset is different from that of the real world, some real fisheye images cannot be well rectified. So we will focus on the image processing based on videos or image sequences to obtain a better rectification effect in the future, and extend our dataset by selecting fisheye images in the real world simultaneously.

References

[1] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *ROBOT AUTON SYST*, vol. 32, no. 1, pp. 1–16, 2000.
[2] Y. Xiong and K. Turkowski, "Creating image-based vr using a self-calibrating fisheye lens," in *CVPR*, 1997.
[3] J. Huang, Z. Chen, D. Ceylan, and H. Jin, "6-dof vr videos with a single 360-camera," in *VIRTUAL REAL-LONDON*, 2017.
[4] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *SIGGRAPH*, 1997.
[5] D. C. Brown, "Close-range camera calibration," *PHOTOGRAMMETRIC ENG*, vol. 37, no. 8, pp. 855–866, 1971.
[6] T. R. Y., "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," in *IEEE J. Robotics and Automation*, 1987.
[7] Z. Zhang, "A flexible new technique for camera calibration," *PAMI*, vol. 22, 2000.
[8] M. D. Grossberg and S. K. Nayar, "A general imaging model and a method for finding its parameters," in *ICCV*, 2001.
[9] P. Sturm and S. Ramalingam, "A generic concept for camera calibration," in *ECCV*, 2004.
[10] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *PAMI*, vol. 28, no. 8, pp. 1335–1340, 2006.
[11] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *ICVS*, 2006.
[12] F. Devernay and O. Faugeras, "Straight lines have to be straight," *MACH VISION APPL*, vol. 13, no. 1, pp. 14–24, 2001.
[13] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca, "Automatic camera calibration applied to medical endoscopy," in *BMVC*, 2009.
[14] R. Melo, M. Antunes, J. P. Barreto, G. Falcao, and N. Goncalves, "Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach," in *ICCV*, 2013.
[15] F. Bukhari and M. N. Dailey, "Automatic radial distortion estimation from a single image," *J MATH IMAGING VIS*, vol. 45, no. 1, pp. 31–45, 2013.
[16] M. Alemán-Flores, L. Alvarez, L. Gomez, and D. Santana-Cedrés, "Automatic lens distortion correction using one-parameter division models," *IPOL*, vol. 4, pp. 327–343, 2014.
[17] M. Zhang, J. Yao, M. Xia, K. Li, Y. Zhang, and Y. Liu, "Line-based multi-label energy optimization for fisheye image rectification and calibration," in *CVPR*, 2015.

[18] J. Rong, S. Huang, Z. Shang, and X. Ying, "Radial lens distortion correction using convolutional neural networks trained with synthesized images," in *ACCV*, 2016.

[19] 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 *ECCV*, 2018.

[20] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma, "Learning to parse wireframes in images of man-made environments," in *CVPR*, 2018.

[21] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *CVPR*, 2017.

[22] Z. Xue, N. Xue, G.-S. Xia, and W. Shen, "Learning to calibrate straight lines for fisheye image rectification," in *CVPR*, 2019.

[23] K. Miyamoto, "Fish eye lens," *JOSA*, vol. 54, no. 8, pp. 1060–1061, 1964.

[24] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern recognition letters*, vol. 16, no. 4, pp. 433–441, 1995.

[25] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *ICVS*, 2006.

[26] C. Toepfer and T. Ehlgen, "A unifying omnidirectional camera model and its applications," in *ICCV*, 2007.

[27] L. Puig, Y. Bastanlar, P. Sturm, J. J. Guerrero, and J. Barreto, "Calibration of central catadioptric cameras using a dlt-like approach," *IJCV*, vol. 93, no. 1, pp. 101–114, 2011.

[28] J. P. Barreto and H. Araujo, "Geometric properties of central catadioptric line images and their application in calibration," *PAMI*, vol. 27, no. 8, pp. 1327–1333, 2005.

[29] R. A. Rensink, "The dynamic representation of scenes," *Vis cogn*, vol. 7, no. 1-3, pp. 17–42, 2000.

[30] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *NeurIPS*, 2014.

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015.

[32] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," *PMLR*, vol. 37, pp. 1462–1471, 07–09 Jul 2015.

[33] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," vol. 97, pp. 7354–7363, 09–15 Jun 2019.

[34] X. Chen, C. Xu, X. Yang, and D. Tao, "Attention-gan for object transfiguration in wild images," in *ECCV*, 2018.

[35] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015.

[36] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *CVPR*, 2017.

[37] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[39] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018.

[40] Blender Online Community, "Blender - a 3d modelling and rendering package," Blender Foundation, Blender Institute Amsterdam, 2014.

[41] A. Eichenseer and A. Kaup, "A data set providing synthetic and real-world fisheye video sequences," in *ICASSP*, 2016.

[42] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IROS*, 2006.