FishFormer: Annulus Slicing-based Transformer for Fisheye Rectification with Efficacy Domain Exploration

Shangrong Yang[1], Chunyu Lin[1*], Kang Liao[1], and Yao Zhao[1]

Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network, Beijing, 100044, China
{sr_yang,cylin,kang_liao,yzhao}@bjtu.edu.cn

Abstract. Numerous significant progress on fisheye image rectification has been achieved through CNN. Nevertheless, constrained by a fixed receptive field, the global distribution and the local symmetry of the distortion have not been fully exploited. To leverage these two characteristics, we introduced Fishformer that processes the fisheye image as a sequence to enhance global and local perception. We tuned the Transformer according to the structural properties of fisheye images. First, the uneven distortion distribution in patches generated by the existing square slicing method confuses the network, resulting in difficult training. Therefore, we propose an annulus slicing method to maintain the consistency of the distortion in each patch, thus perceiving the distortion distribution well. Second, we analyze that different distortion parameters have their own efficacy domains. Hence, the perception of the local area is as important as the global, but Transformer has a weakness for local texture perception. Therefore, we propose a novel layer attention mechanism to enhance the local perception and texture transfer. Our network simultaneously implements global perception and focused local perception decided by the different parameters. Extensive experiments demonstrate that our method provides superior performance compared with state-of-the-art methods.

Keywords: Fisheye Image Rectification, Transformer, Global and Local Perception, Annulus Slicing

1 Introduction

Benefiting from the large field of view (FOV), the fisheye camera has become popular in several computer vision tasks such as robot navigation [1], object detection and tracking [2], and motion estimation [3]. Different from the ideal pinhole model, fisheye cameras distort the incident light to accommodate more content in limited image space, thus they distort the structure as well. The distortion dramatically affects many computer vision algorithms designed on perspective images. Therefore, it is necessary to correct the distortion of the fisheye image.

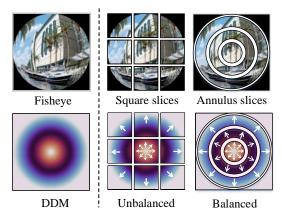* Corresponding author: cylin@bjtu.edu.cn

Fig. 1.   Given a distortion distribution map (DDM), existing square slicing method (middle) complicates the distortion distribution between patches, thereby increasing the difficulty of network learning. Our learning conducive annulus slicing method (right) ensures the distortion uniformity in a patch and distortion gradualness between different patches.

The distortion in the fisheye image primarily affects the image structure rather than content. Therefore, most previous methods used significant distortion features, such as conics or distorted objects, to help distortion correction. The manual calibration methods [4][5] can effectively find conics, but it is time-consuming. Self-calibration methods [6][7][8] replace artificially searching with automated algorithms, but the found features are not always reliable. To solve these problems, many studies rely on deep neural networks. The regression-based method [9][10][11][12] simplifies the distortion correction to a parameter-prediction problem. Generation-based method [13][14] directly leverages the generative adversarial network (GAN) to achieve image-to-image correction. Similarly, they use various features, such as edges [14], optical flow [15] to enhance the quality of generation. However, these methods have two issues: (1) Convolving the entire image straightforward with unified perception is not conducive to perceiving the global distortion. (2) The radial symmetry of fisheye images is not fully exploited to improve prediction accuracy.

To handle the problems above, we propose our Fishformer. We improved the Transformer according to the structural characterizations of fisheye images. Note that distortion in fisheye images has high symmetry. But the existing square slicing method would break this symmetry. As shown in Figure 1, it will complicate the range of distortion between patches, thereby increasing the difficulty of global perception. Therefore, we introduce a learning conducive annulus slicing method. The fisheye image is sliced in the annulus to ensure distortion uniformity, facilitating global prediction. By visualizing the efficacy domains of the fisheye parameters, we found different distortion parameters affect relatively fixed regions in an image. *This phenomenon implies that the perception of the local area can further improve the prediction accuracy.* Therefore, we propose a novel layer attention mechanism, which calculates the attention of features between adjacent

layers. In this way, the rich texture in the first layer feature will be continuously transferred, thereby enhancing the perception of local areas. Since each patch is in a different area, the confidence of the distortion parameters predicted by each patch should be related to their location. In order to implement focused local perception of different parameters on different regions, we use a set of rough truncated normal distribution probability densities as the weight of patch prediction loss.

To summarize, we make the following contributions:

– We process fisheye images as sequences for the first time and propose a learning conducive annulus slicing method to ensure the uniformity and graduality of regional distortion.
– Based on discovered parameters efficacy domains, we propose a novel layer attention mechanism to promote the texture transmission and enhance the focused local perception in different regions.
– Our network achieves the perception of global and local texture simultaneously. Extensive experiments on different datasets demonstrate the superiority of our method.

2   Related Work

Distortion will drastically decrease the performance of many perspective image-based algorithms. The distortion rectification can effectively alleviate this problem and thus receive significant attention. The earliest attempt used traditional machine learning methods [16][17][18][19] to exploit distortion correction. Mei et al. [16] proposed a method to calculate the radial distortion parameter using a calibration board with coplanar points. It can accurately calculate the parameters but require special equipment and manual participation. Self-calibration methods [20][21][22][23] broke down the limitation of human participation. Zhang et al. [20] designed a method following the rule that straight lines have to be straight [24]. *Detecting the edges and calculating the distortion parameters.* Although the automatic calibration method was more convenient, the detected features were easily affected by the image content, resulting in unstable performance.

With the continued development of deep learning, many researchers considered using deep convolutional networks to correct distortion. Rong et al. [10] regard distortion correction as a regression task and first introduced Alexnet to perceive distortion and predict parameters. Since their modeling of the fisheye image is straightforward and the parameter is in a relatively small range, the rectification is limited. Yin et al. [11] proposed a FisheyeRecnet, which is aided by the semantic information to perceive the distortion better. However, the pre-training of the semantic segmentation network further increased the difficulty of training. Similarly, Xue et al. [12] enhanced their regression network by leveraging the edges detector, which also needs to be pre-trained.

Liao et al. [13] first considered distortion correction as a generation task. They introduced a simple generative adversarial network (GAN) to learn the
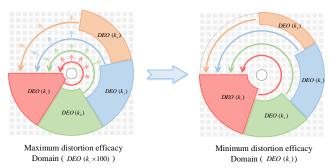
Fig. 2.   **Distortion efficacy domain (DEO).** The visualization of the maximum (left) and minimum (right) distortion efficacy domain. The area of the DEO will increase with the value of k and $DEO\,(k_1) \geqslant DEO\,(k_2) \geqslant DEO\,(k_3) \geqslant DEO\,(k_4)$.

distribution difference and achieved one-stage correction. However, naive GAN cannot learn the distribution difference well, so the corrected image has a noticeable artifact. DDM [14] improved the correction accuracy by distortion distribution map, but it cannot prevent the content of the corrected image from being changed. Based on this, Yang et al. [25] believed that the issue of the ambiguity resides in the transmitted distorted structure, so a network is introduced to isolate the distortion transmission. Although the method of Yang et al. [25] has a significant improvement, the accuracy of the appearance flow produced by the self-supervised method is a challenge.

3   Fisheye Distortion

3.1   Distortion Model

Since large amount of real fisheye images and their corresponding distortion parameters are difficult to obtain. Using distortion models to synthesize fisheye images has become the mainstream choice. Generally, the commonly used fisheye distortion models is polynomial model [26]. Suppose the coordinate of an arbitrary point on normal image is $p\,(x,y)$. Its corresponding coordinate on fisheye image is $p'\,(x',y')$. The Euclidean distance r_u from $p\,(x,y)$ to the image center $c\,(x_0,y_0)$ corresponds to the Euclidean distance r_d from p_d to the distortion center $c_d\,(x_d,y_d)$. They can be expressed as:

$$r_u = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{1}$$

$$r_d = \sqrt{(x' - x_d)^2 + (y' - y_d)^2} \tag{2}$$

The polynomial model uses high-order polynomials to fit the complex distortion on fisheye image and can be denoted as:

$$x = (1 + k_1 r_d^2 + k_2 r_d^4 + \cdots)x' \tag{3}$$

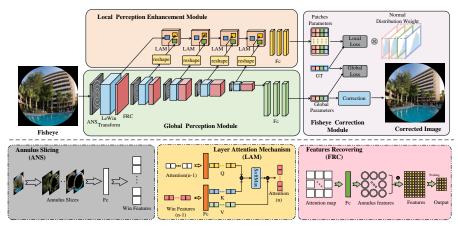$$y = (1 + k_1 r_d^2 + k_2 r_d^4 + \cdots)y' \tag{4}$$

Fig. 3. The network architecture of our Fishformer, mainly consists of a global perception module and a local perception enhancement module. In the global perception module, the features are sliced with the annulus slicing method (ANS), then leveraging the LeWin Transformer [27] to calculate the attention. The attention features are recovered (FRC) *to a complete feature.* In the local perception enhancement module, each layer feature is used in layer attention mechanism (LAM) to enhance local perception further.

Merge the above formulas when the distortion center $c_d\,(x_d, y_d)$ is the image center. The polynomial model used to describe the relationship between r_u and r_d can be obtained:

$$r_u = (1 + \sum_{i=1}^{n} k_i r_d{}^{2i}) r_d \tag{5}$$

3.2 Distortion efficacy domain

Higher-order polynomials can accurately model real fisheye images. However, the difficulty of network prediction will increase with the number of parameters. Usually, large amount of previous research [11] [13] [12] considered that the four parameters are appropriate choice. Therefore, we also select a four-parameter polynomial model. In addition, for intuitively observing the influence of different parameters on variable areas on the fisheye image, we simplify the polynomial model by treating the image center as the distortion center:

$$r_u = (1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + k_4 r_d^8) r_d \tag{6}$$

Notice that the polynomial multiplied by r_d contains a linear constant 1 and a nonlinear term $\sum_{i=1}^{4} k_i r_d{}^{2i}$. The linear constant represents the scaling transform and does not have impact on distortion. Nonlinear term $\sum_{i=1}^{n} k_i r_d{}^{2i}$ denotes non-linear transform. To explore the efficacy domain of each nonlinear

term, we decompose the formula:

$$r_u = \left(\frac{1}{4} + k_1 r_d{}^2\right) r_d + \left(\frac{1}{4} + k_2 r_d{}^4\right) r_d +$$
$$\left(\frac{1}{4} + k_3 r_d{}^6\right) r_d + \left(\frac{1}{4} + k_4 r_d{}^8\right) r_d \tag{7}$$

We set $r(k_i) = \left(\frac{1}{4} + k_i r_d{}^{2i}\right) r_d$. Each $r(k_i)$ is only affected by a linear term and a nonlinear term, and each nonlinear term has only one k. To explore the area where each k value works, we only keep $r(k_i)$ and set the rest to 0. In this way, there are four independent situations:

$$k_1 \epsilon \left[10^{-6}, 10^{-4}\right], r_u = \left(\frac{1}{4} + k_1 r_d{}^2\right) r_d$$
$$k_2 \epsilon \left[10^{-11}, 10^{-9}\right], r_u = \left(\frac{1}{4} + k_2 r_d{}^4\right) r_d$$
$$k_3 \epsilon \left[10^{-16}, 10^{-14}\right], r_u = \left(\frac{1}{4} + k_3 r_d{}^6\right) r_d \tag{8}$$
$$k_4 \epsilon \left[10^{-21}, 10^{-19}\right], r_u = \left(\frac{1}{4} + k_4 r_d{}^8\right) r_d$$

Among them, please refers to [13] [12] for the range of $[k_1, k_2, k_3, k_4]$. We use the above four formulas to synthesize fisheye images independently. Then, we provide formula $r_u - \frac{1}{4} r_d < 1$ to divide the distorted region and the distortion-negligible region. The distorted area is called the distortion efficacy domain(DEO). We draw the DEO of each parameter according to the value range of Equation (8). The DEO of each parameter appears as a 360-degree annular (in Figure 2 we only color a part of the annular). When we increase the value of k_i, its corresponding DEO area will increase. The DEO with higher-order coefficients is always smaller than the DEO with lower-order coefficients. The relationship can be expressed as:

$$DEO\left((k_i)_{max}\right) \geqslant DEO\left(k_i\right) \geqslant DEO\left((k_i)_{min}\right) \tag{9}$$

$DEO\left(\cdot\right)$ is the area of the distortion efficacy domain. In the same quantization interval, the higher order of k has the smaller area of the domain. It can be expressed as:

$$K_N = \sum_i^n k_{N\_i} \tag{10}$$

$$DEO\left(k_{1\_i}\right) \geqslant DEO\left(k_{2\_i}\right) \geqslant$$
$$DEO\left(k_{3\_i}\right) \geqslant DEO\left(k_{4\_i}\right) \tag{11}$$

K_N is the range of the N-th $k, N \epsilon [1, 2, 3, 4]$, k_{N_i} is the i-th interval of K_N. In this way, we can determine the approximate efficacy domains of each k, then we design a network that can focus on different parameters on different regions.

4   Proposed Method

Most of the existing deep learning methods use convolution to perform local perception but fail to exploit the prior information better. In fact, the distortion structure of the fisheye image has good symmetry. The degree of distortion increase with the radius, showing good progress. Therefore, we process the fisheye image as a sequence. We design our Fishformer based on the existing popular LeWin Transformer [27], as shown in Figure 3. Our network comprises a global perception module and a local enhanced perception module. We first slice the fisheye image in the annulus and send them to the global perception module for feature extraction and parameter prediction. Subsequently, we send the extracted features in each layer to the local enhanced perception module. Calculating the attention between two adjacent layer features maintains local texture transfer and reinforces the local perception.

4.1   Global Perception Module

Annulus Slicing (ANS). The global perception module mainly uses five Fishformer blocks to extract the features and make a global prediction. In the Transformer block, different from the previous methods, we first use equidistant annulus slicing with a width of 8 to replace the traditional square slicing. Square slicing destroys this regularity, leading to inconsistent distortions between patches, thereby increasing the difficulty of network training. In contrast, annulus slicing allows each patch to maintain the same characteristics. The distortion degree in the same patch still remains radial symmetry, and the distortion degree between the different patches maintains an orderly progression. Therefore, the network can learn the distortion characteristics easily. Assuming that the number of slices is n, we can transform an image feature of shape $(1, w, h, c)$ into (n, w, h, c) by annulus slicing. Note that annulus slicing causes redundant areas without valid pixels in each patch. Thus, we transform slices into vectors with shape $(n, w*h, c)$ and feed them into a fully connected layer to extract the effective pixels. We get the output with shape $(n, 8*8, c)$ and arrange it into an 8×8 patch window with shape $(n, 8, 8, c)$.

Features Recovering (FRC). Assisted by the LeWin Transformer structure [27], each of our Fishformer blocks contains two LeWin Transformer layers to calculate the patch attention. After obtaining the attention with shape $(n, 8, 8, c)$, we need to introduce a fully connected layer for inverse transformation, turning the attention result into the annulus shape with size of (n, w, h, c) and adding them to generate the final attention feature with shape $(1, w, h, c)$. We perform a down-sampling for the attention feature and get the input with shape $(1, w/2, h/2, c)$ for the next block. After five Fishformer blocks, three fully connected layers are used to predict global distortion parameters P_{gb}.

We leverage L1 distance to supervise. The distortion parameter ground truth is denoted as P_{gt} and the loss function can be expressed as

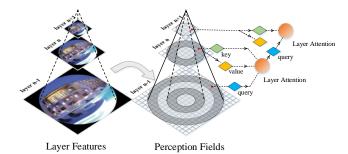$$\mathcal{L}_{global} = \sum_{i=1}^{4} \|P_{gb} - P_{gt}\|_1 \tag{12}$$

Fig. 4. **Layer attention mechanism.** By calculating the attention between layer features, the rich texture is continuously passed forward and the relationship between adjacent patches can be quickly constructed.

4.2   Local Perception Enhancement Module

Layer Attention Mechanism (LAM). Efficacy domains indicate that different parameters influence different areas. Therefore, the local perception is as important as the global perception. However, Transformer generally focuses on perceiving long-range information and neglecting the local context. Besides, the LeWin Transformer structure down-samples the generated attention features, which will cause the loss of local texture. To solve these problems, we introduce a novel layer attention mechanism. We use the window patches with a size of $(n, 8, 8, c)$ in the first transformer block as query, and the window patches in the next layer with shape $(n/2, 8, 8, c*2)$ as key and value to calculate the attention map. Noticing that the shapes of query and key are different, we resize key and value to the shape of query as the shape $(n, 8, 8, c)$ to calculate the attention. After obtaining the attention map, we resize it to $(n/2, 8, 8, c*2)$. Leveraging it as a new query and the subsequent window patches with shape $(n/4, 8, 8, c*4)$ as key and value, we can calculate the new attention map. In this way, the intense texture detail in the first Transformer block can be continuously forward passed. Compared with LeWin Transformer, the layer attention mechanism can fully exploit the relationship between layer features. The relationship between adjacent patches can be quickly constructed and achieve progressive perception, as shown in Figure 4. Our purpose is to enhance the local perception. Therefore, for each patch, the network needs to predict corresponding distortion parameters. We feed the final attention map into three fully connected layers and obtain multiple patches parameters.

Loss Function of Patch Parameters based on Normal Distribution Confidence. In the local perception enhancement module, the network locally perceives each patch and then predicts their corresponding distortion parameters. However, each patch of the fisheye image is variably affected by distortion parameters. It means that perceiving different patches assists the network to produce an accurate prediction for some of the parameters. Considering these characteristics, we introduce different truncated normal distribution densities for different parameters as their confidence. There are four truncated normal distribution density functions for our four distortion parameters. We artificially

set their variance σ to be the same, denoted as $\sigma = 1$. Their mean μ is set to $[-1, -0.5, 0.5, 1]$ and truncated interval is set to $[-2, 2]$. The normal distribution density function and parameters confidence is as follows:

$$f\left(x, \sigma, \mu\right) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{13}$$

$$C\left(k_i\right) = f\left(x, 1, \mu_i\right), \mu_i \epsilon\left[-1, -0.5, 0.5, 1\right] \tag{14}$$

After obtaining the weight of the patch predicted parameters $C_t\left(k_i\right)$, we calculate a weighting L2 distance for patch predicted parameters P_{pt} and ground truth P_{gt} as the loss of local enhancement perception module. The process can be expressed as:

$$\mathcal{L}_{local} = \sum_{i=1}^{4} \sqrt{\frac{\sum_{t=1}^{n} C_t\left(k_i\right)\left[P_{pt}\left(k_i\right) - P_{gt}\left(k_i\right)\right]^2}{\sum_{t=1}^{n} C_t\left(k_i\right)}} \tag{15}$$

Where n is the number of patches.

4.3   Fisheye Correction Module

After obtaining the predicted distortion parameters, the network has been able to achieve regular training. However, it is not a complete end-to-end process from the fisheye image to the corrected image. Therefore, we use the predicted distortion parameters to pre-correct the fisheye image.

The pixels of the fisheye image and the standard image are not a one-to-one correspondence. There are many holes in the pre-corrected image. We use a moving average filtering method to fill in missing pixels. Specifically, we first calculate a mask M from the pre-corrected image I_p. We leverage moving average filtering on the pre-corrected image to generate a blurred corrected image I_b. Then we take out the filled pixels through a mask and add them to the pre-corrected image to get the final corrected image I_c. The inpainting process can be expressed by the following formula:

$$I_c = M \cdot I_p + (1 - M) \cdot I_b \tag{16}$$

After obtaining the corrected image, we calculate the image Loss to help the network training. There are two advantages: (1)The gap between the image domains is relatively small, which benefits the network to estimate parameters quickly. (2)A complete end-to-end training from fisheye image to corrected image is realized.

We use L2 distance as the image loss to supervise the quality of the corrected image. The calculation can be expressed as:

$$\mathcal{L}_{image} = \|I_c - I_{gt}\|_2 \tag{17}$$

Finally, we construct our training loss function by summing together all losses as follows:

$$\mathcal{L} = \mathcal{L}_{global} + \mathcal{L}_{local} + \mathcal{L}_{image} \tag{18}$$

Table 1. Comparison between the proposed method and the state-of-the-art.

Method	Places2					CelebA				
	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
Blind [15]	14.71	0.4716	0.5487	189.5	0.6779	17.19	0.6101	0.6645	119.4	0.7759
DCCNN [10]	15.17	0.4760	0.3668	190.8	0.6874	15.46	0.5697	0.4311	66.4	0.7193
DRGAN [13]	17.75	0.5558	0.7174	164.9	0.7586	18.22	0.6551	0.7563	129.9	0.8022
RPC [31]	19.74	0.5497	0.6323	167.6	0.7687	19.47	0.6854	0.7332	75.7	0.8228
DeepCalib [30]	20.84	0.6868	0.7722	69.7	0.8537	18.03	0.7138	0.7449	140.1	0.8177
DDM [14]	24.69	0.7952	0.9173	79.5	0.9205	26.53	0.8746	0.9317	52.0	0.9374
PCN [25]	25.10	0.8152	0.9178	**65.8**	0.9323	27.34	0.8844	0.9466	39.2	0.9534
Ours(FishFormer)	**25.43**	**0.8412**	**0.9411**	73.4	**0.9388**	**27.62**	**0.9071**	**0.9595**	**22.3**	**0.9566**

Table 2. Cross-testing between our method and a part of the-state-of-art methods.

Method	Places2 → CelebA					CelebA → Places2				
	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
Blind [15]	14.89	0.5742	0.6266	171.5	0.7439	15.93	0.4800	0.5686	222.9	0.7048
DCCNN [10]	14.91	0.5553	0.3978	76.9	0.7051	15.31	0.4792	0.3757	186.2	0.6913
DRGAN [13]	17.61	0.5921	0.7281	127.1	0.7771	15.13	0.4672	0.5501	298.6	0.6765
PCN [25]	27.68	0.8963	0.9523	25.3	0.9595	22.57	0.7235	0.8377	111.4	0.8866
Ours(Fishformer)	**27.89**	**0.9185**	**0.9685**	**22.4**	**0.9614**	**23.00**	**0.7509**	**0.8546**	**81.6**	**0.8953**

5   Experiments

5.1   Dataset Generation and Experiment Details

We select two completely different types of datasets to verify the effectiveness of our method. (1)Place2 dataset [28], which contains more than 10 million images, covering 400 indoor and wild scenes. (2)CelebA dataset [29], which includes 2 million face images. We randomly select 44k images from these two datasets as our original images. Since our method requires annulus slicing according to the distortion distribution, and most real fisheye images are circular. Therefore, we crop the image to be circular instead of square in previous methods [15][10][13][30][14][25]. We leverage the four-parameters polynomial model to transform the normal image, generating fisheye images with 128×128 resolution. The value of each parameter is randomly selected in the range described in Equation (8) for data augmentation. 40K synthetic fisheye images are randomly selected for training, leaving 4k images for testing. For the Place2 and CelebA datasets, we train two different models separately. Considering a robust rectification model, it should not attach *extreme* to the images content but focus on the structure. We thereby exchange the training and test on the Place2 and the CelebA. For example, we evaluate the training model from Places2 on the CelebA test set to verify our network performance and vice versa.

5.2   Quantitative Comparison

To measure our method performance, we compared with state-of-the-art methods including Blind [15], DCCNN [10], DRGAN [13], RPC [31], DeepCalib [30], DDM [14], PCN [25]. Deep-learning methods [15][10][13][31][14][25] are retrained on our circular dataset except [30]. For DeepCalib [30] with a special model, panoramic images are used to generate fisheye images and corresponding ground truth. However, our dataset has no corresponding panoramic images. Therefore, we did not retrain the DeepCalib [30]. We use its regression pretrained model to test our circular images. To improve fairness, we scale the corrected
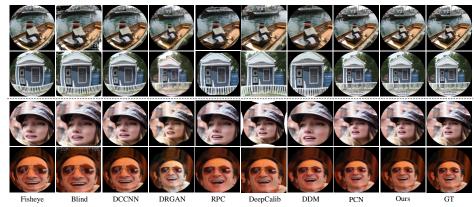
| Fisheye | Blind | DCCNN | DRGAN | RPC | DeepCalib | DDM | PCN | Ours | GT |

Fig. 5.   Qualitative Comparison results. From left to right, we demonstrate the rectification results of Blind [15], DCCNN [10], DRGAN [13], RPC [31], DeepCalib [30], DDM [14], PCN [25], and our method. Our comparative experiments are tested on synthetic Places2 dataset [28] (top two rows) and CelebA [29] dataset (bottom two rows), respectively.
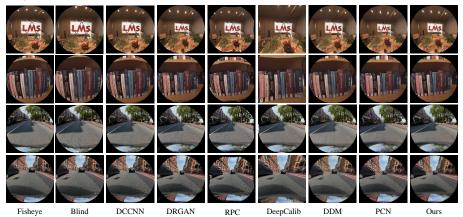

| Fisheye | Blind | DCCNN | DRGAN | RPC | DeepCalib | DDM | PCN | Ours |

Fig. 6.   Rectification for real fisheye images. We test the state-of-the-art methods on different real fisheye datasets, including LMS fisheye dataset [35] (top) and Woodscape fisheye dataset [36] (bottom).

images with different radius as candidate results. Calculating the performance for candidate results, the maximum value is taken as the final performance.

We leverage PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity), MS-SSIM (Multiscale structural similarity) [32], FID (Frechet Inception Distance) [33], and CW-SSIM (Complex wavelet structural similarity) [34] to evaluate the performance.

We test on the Place2 dataset and CelebA dataset respectively, and also conduct cross-testing. The experiment results are shown in Table 1. In general, the test results on the CelebA [29] are better than those on the Places2 [28]. The complexity of the images on CelebA [29] is smaller than that on Places2 [28], so it is easier to correct. Limited by the simple model, the performance of
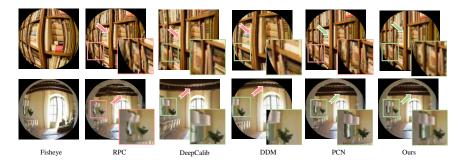
Fig. 7. Further comparison in rectification quality. The arrows and rectangular boxes highlight the corrected structure and content, respectively. Colors represent different qualities: reasonable quality (green), and poor quality (pink).

[15][10][13] is worse than [31]. [30] is better than [31], because its model is closer to the real fisheye images. [14] and [25] utilize Generative Adversarial Networks (GAN) to assist correction with more features, so they lead to a more significant performance boost. As can be seen, the PCN [25] is the best deep learning method, but the performance is still lower than our method. When exchanging training models for testing, the model trained on Places2 [28] can achieve better performance on CelebA [29]. In contrast, the performance will drop when testing the model trained on CelebA. This phenomenon is because the data complexity of Places2 [28] is greater than that of CelebA [29], so the trained model can be easily applied to low-complexity data. Nevertheless, our approach is still better than PCN [25], as shown in Table 2.

5.3   Qualitative Comparison

We visualized the calibration results of all comparison methods and our method. The synthetic images corrected results are shown in Figure 5. The real fisheye images rectified results including LMS dataset [35] and Woodscape dataset [36] are shown in Figure 6.

The correction effect of [15][10] are not significant because their model is relatively simple. [13] can correct the structure well, but it will produce obvious artifacts. The two-stage correction of [31] further reduce the artifacts. However, the structure correction is still inaccurate. The structural correction of [30] is accurate, while the effect varies according to the different images. [14] and [25] are very accurate in structure correction. Nevertheless, there are some local artifacts caused by the characteristics of the GAN. In general, the generation-based method [15][13][14][25][31] is better in structure correction, but the generated images have an artifact to some extent. Compared with the generation-based method, the regression method [10][30] only needs to warp and interpolate the original image according to the predicted parameters. There is less artifact in the central area for most preserved original pixels. Methods [10][30] do not have the problem of artifact content, but the structure correction is deficient. In contrast, our corrected results achieve quite accurate structure, as shown in Figure 7.

Table 3. Result of using different slicing methods and different number of patches.

Square/Annulus	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
1	22.96/22.96	0.7440/0.7440	0.8539/0.8539	81.7/81.7	0.8933/0.8933
4	23.41/23.81	0.7568/0.7786	0.8711/0.8891	82.3/82.1	0.9031/0.9112
16	24.11/**25.34**	0.7941/**0.8314**	0.9023/**0.9326**	79.8/**74.1**	0.9176/**0.9353**
64	24.22/24.75	0.7914/0.8149	0.8975/0.9224	79.6/78.9	0.9166/0.9271

Table 4. Performance of using layer attention mechanism.

	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
W LTM	**25.43**	**0.8412**	**0.9411**	**73.4**	**0.9388**
W/O LTM	25.34	0.8314	0.9326	74.1	0.9353

5.4   Ablation study

To further analyze our network, we introduce many additional experiments to exploit the improvements from different key components, including the slicing method, combination of layer attention mechanism, and the network depth.

Slicing method and number of patches To prove the effectiveness of our proposed slicing method, we use traditional square slicing and annulus slicing to conduct experiments. Besides, we also verify the impact of the patches number. Except for the slicing method and the number of patches, we keep the remaining settings consistent. The experiment results are shown in Table 3.

The performance without using the slicing method (patch is 1) is worse than using the slicing method. Compared with square slicing, the performance of annulus slicing is always better regardless of the number of patches. This performance effectively proves that annulus slicing can benefit the network to learn the distortion distribution. By using annulus slicing, as the patches number increases, the network achieves optimal performance when the number is 16. It demonstrates that the patches number is not the more the better. Training difficulty will increase with the patches number.

Combination of layer attention mechanism In this paper, we propose a novel layer attention mechanism. To verify its effectiveness, we removed it and only used the global perception module to predict the distortion parameters. The results are shown in Table 4. It can be seen that our layer attention mechanism can bring certain improvements.

Subsequently, we explored the combination of layer attention mechanism. By exploring the combination of q, k, v, we search for optimal performance. The search results are shown in Table 5. We found that when the first layer features are used as q, the obtained performance is the best. This combination can retain more local texture so that the network pays more attention to local perception.

In addition, to explore whether the local perception is necessary participate in global perception, we deliver the feature in the local perception enhancement module to the global perception module on each layer (*w deliver*) and get the experiment results in Table 6. The improvement is not apparent. We also conduct additional exploration on layer attention mechanism. Using skip layer features

Table 5. Performance of different combinations.

Layer I	Layers II-V	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
q	kv	25.43	**0.8412**	**0.9411**	**73.4**	**0.9388**
k	qv	**25.53**	0.8396	0.9386	74.0	0.9384
v	qk	25.26	0.8279	0.9309	75.1	0.9338
kv	q	25.23	0.8334	0.9358	75.2	0.9347
qv	k	25.08	0.8281	0.9302	74.8	0.9329
qk	v	25.09	0.8282	0.9302	74.9	0.9329

Table 6. Performance of way of layer attention mechanism.

	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
w deliver	25.43	0.8412	0.9404	**72.2**	0.9376
w/o deliver	**25.43**	**0.8412**	**0.9411**	73.4	**0.9388**
w skip	25.43	**0.8412**	**0.9411**	**73.4**	**0.9388**
w/o skip	**25.45**	0.8378	0.9384	73.7	0.9376

Table 7. Performance of different depth.

num	PSNR ↑	SSIM ↑	MS-SSIM ↑	FID ↓	CW-SSIM ↑
3	22.98	0.7485	0.8545	82.2	0.8943
4	25.06	0.8208	0.9256	75.4	0.9306
5	**25.34**	**0.8314**	**0.9326**	**74.1**	**0.9353**
6	25.14	0.8294	0.9322	75.9	0.9337
7	24.53	0.8038	0.9124	77.9	0.9246

(*w skip*) and layer by layer features (*w/o skip*) to calculate attention separately. The results are shown in Table 6. The layer-by-layer (*w/o skip*) result is better than the skip layer (*w skip*) result. It demonstrates that layer-by-layer attention can perceive local texture more meticulously.

The network depth Finally, we explored the influence of the Transformer blocks number. The results are shown in Table 7. When the number of blocks is 5, the performance is optimal. To a certain extent, the performance improvement is not apparent when increasing the block number. This phenomenon means that increasing the number of blocks does not improve performance indefinitely.

6   Conclusion

In this paper, we design a network for correcting fisheye images with the help of Transformer structure, processing the fisheye images as a sequence for the first time. Considering the Transformer slicing method is not appropriate, an annulus slicing that conforms to the distortion distribution of the fisheye image is proposed. According to the observation of the parameters, we found the distortion efficacy domain and designed a novel layer attention mechanism to enhance the local texture perception. For the problem of inconsistency confidence of the different patch parameters, we introduce a set of truncated normal

distribution probability densities as the weight of patch loss to make the training more reasonable. Our network has a structure that combines global and local perceptions. The test results on two completely different datasets(Place2 and CelebA) prove the effectiveness of our method. Cross-testing between models further verifies that our approach has a better performance.

References

1. J.Greer, L.Blumenschein, R.Alterovitz, E.Hawkes, and A.Okamura. Robust navigation of a soft growing robot by exploiting contact with the environment. *The International Journal of Robotics Research*, 39(14):1724–1738, Mar 2020.
2. S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46:1772–1788, 2013.
3. T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.
4. P. Reimer, E. Bard, A. Bayliss, J. Beck, P. Blackwell, C. Ramsey, C. Buck, Hai Cheng, R. Edwards, M. Friedrich, P. Grootes, T. P. Guilderson, H. Haflidason, I. Hajdas, C. Hatté, T. Heaton, D. Hoffmann, A. Hogg, K. Hughen, K. Kaiser, B. Kromer, S. Manning, M. Niu, R. Reimer, D. Richards, E. M. Scott, J. Southon, S. Richard, C. Turney, and J. D. Plicht. Intcal and marine radiocarbon age calibration curves 50 , 000 years cal bp. 2013.
5. Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 1:666–673 vol.1, 1999.
6. D. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1027–1034, 2013.
7. M. Stuiver and P. Reimer. Extended 14c data base and revised calib 3.0 14c age calibration program. *Radiocarbon*, 35:215–230, 1993.
8. R. Melo, M. Antunes, J. Pedro Barreto, G. Falcão Paiva Fernandes, and N. Gonçalves. Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach. *ICCV*, pages 537–544, 2013.
9. A. Krizhevsky, S. Ilya, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *CACM*, 2017.
10. J. Rong, S. Huang, Z. Shang, and X. Ying. Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *ACCV*, 2016.
11. 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*, pages 475–490, 2018.
12. Z. Xue, N., G. Xia, and W. Shen. Learning to calibrate straight lines for fisheye image rectification. *CVPR*, pages 1643–1651, 2019.
13. K. Liao, C. Lin, Y. Zhao, and M. Gabbouj. DR-GAN: Automatic radial distortion rectification using conditional GAN in real-time. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
14. K. Liao, C. Lin, Y. Zhao, and M. Xu. Model-free distortion rectification framework bridged by distortion distribution map. *IEEE Transactions on Image Processing*, 29:3707–3718, 2020.
15. X. Li, B. Zhang, Pedro V. Sander, and J. Liao. Blind geometric distortion correction on images through deep learning. In *CVPR*, pages 4855–4864, 2019.

16. C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. *IEEE International Conference on Robotics and Automation*, pages 3945–3950, 2007.
17. S. Gasparini, P. F. Sturm, and J. Pedro Barreto. Plane-based calibration of central catadioptric cameras. *ICCV*, pages 1195–1202, 2009.
18. L. Puig, Y. Bastanlar, P. Sturm, J. J. Guerrero, and J. Barreto. Calibration of central catadioptric cameras using a dlt-like approach. *International Journal of Computer Vision*, 93:101–114, 2010.
19. Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1330–1334, 2000.
20. 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. *CVPR*, pages 4137–4145, 2015.
21. J. Pedro Barreto and H. Sabino de Araújo. Geometric properties of central catadioptric line images and their application in calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1327–1333, 2005.
22. G. Chander, B. Markham, and D. Helder. Summary of current radiometric calibration coefficients for landsat MSS, TM, ETM+, and EO-1 ALI sensors. *Remote Sensing of Environment*, 113:893–903, 2009.
23. G. Andreas, F. Moosmann, C. Omer, and B. Schuster. Automatic camera and range sensor calibration using a single shot. *2012 IEEE International Conference on Robotics and Automation*, pages 3936–3943, 2012.
24. F. Devernay and O. Faugeras. Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments. *Machine Vision Applications*, 13(1):14–24, 2001.
25. S.Yang, C.Lin, K.Liao, C.Zhang, and Y.Zhao. Progressively complementary network for fisheye image rectification using appearance flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6348–6357, 2021.
26. A. Basu and S. Licardie. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16:433–441, 1995.
27. Z.Wang, X.Cun, J.Bao, and J.Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021.
28. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018.
29. Z.Liu, P.Luo, X.Wang, and X.Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
30. O. Bogdan, V. Eckstein, F. Rameau, and J. Bazin. Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *CVMP*, 2018.
31. K.Zhao, C.Lin, K.Liao, S.Yang, and Y.Zhao. Revisiting radial distortion rectification in polar-coordinates: A new and efficient learning perspective. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2021.
32. Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multi-scale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402, 2003.
33. H. Martin, R. Hubert, U. Thomas, N. Bernhard, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

34. M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, and M. K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Transactions on Image Processing*, 18:2385–2401, 2009.
35. A.Eichenseer and A.Kaup. A data set providing synthetic and real-world fisheye video sequences. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1541–1545, 2016.
36. S.Yogamani, C.Witt, H.Rashed, S.Nayak, S.Mansoor, P.Varley, X.Perrotton, D.Odea, P.Perez, C.Hughes, J.Horgan, G.Sistu, S.Chennupati, M.Uricar, S.Milz, M.Simon, and K.Amende. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9308–9318, 2019.