Progressively Complementary Network for Fisheye Image Rectification Using Appearance Flow

Shangrong Yang*, Chunyu Lin*†, Kang Liao*, Chunjie Zhang, Yao Zhao
Institute of Information Science, Beijing Jiaotong University
Beijing Key Laboratory of Advanced Information Science and Network, Beijing, 100044, China
{sr_yang, cylin, kang_liao, cjzhang, yzhao}@bjtu.edu.cn

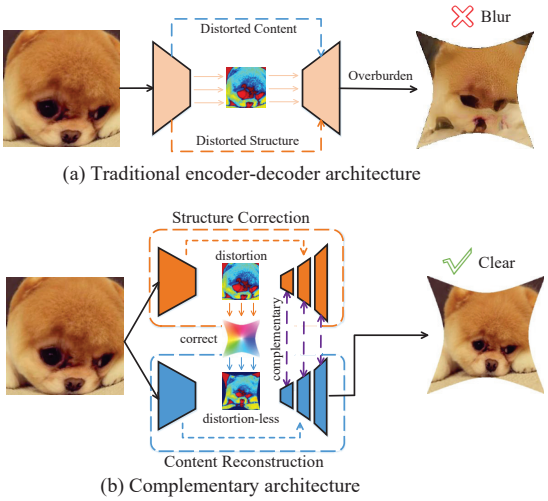
Abstract

Distortion rectification is often required for fisheye images. The generation-based method is one mainstream solution due to its label-free property, but its naive skip-connection and overburdened decoder will cause blur and incomplete correction. First, the skip-connection directly transfers the image features, which may introduce distortion and cause incomplete correction. Second, the decoder is overburdened during simultaneously reconstructing the content and structure of the image, resulting in vague performance. To solve these two problems, in this paper, we focus on the interpretable correction mechanism of the distortion rectification network and propose a feature-level correction scheme. We embed a correction layer in skip-connection and leverage the appearance flows in different layers to pre-correct the image features. Consequently, the decoder can easily reconstruct a plausible result with the remaining distortion-less information. In addition, we propose a parallel complementary structure. It effectively reduces the burden of the decoder by separating content reconstruction and structure correction. Subjective and objective experiment results on different datasets demonstrate the superiority of our method.

1. Introduction

Currently, fisheye cameras are widely used in video surveillance [1], autonomous driving [2] and mobile applications [3]. However, the images captured by fisheye cameras are not suitable for most computer vision techniques designed for perspective images, such as target tracking [4][5], motion estimation [6][7], scene segmentation [8][9]. In order to resolve the contradiction, distortion rectification has drawn great attention for decades.

Traditional algorithms [10][11][12][13] automatically extract pervasive features to calculate corresponding model parameters. However, the number of detected features is unstable, which greatly influences the model accuracy. To solve this problem, many existing approaches [14][15][16][17][18][19] leverage the potential of deep learning which can roughly be divided into two categories: regression-based method and generation-based method. The regression-based methods [14][15][16][17] utilize convolutional neural network (CNN) to predict complex nonlinear model parameters. However, they have to trade-off the number of parameters in the nonlinear model. In contrast, the generation-based methods [18][19][20] directly generate corrected images with the help of encoder-decoder structure. Nevertheless, the effects inferred by this structure have never been explored. As we can see from Fig. 1a, the skip-connection in structure communicates redundant information, such as distorted features extracted by the encoder, thus confounding the decoder. The transmitted distorted features present difficulties for image reconstruction,



(a) Traditional encoder-decoder architecture

(b) Complementary architecture

Figure 1. **Generation-based approaches for image rectification.** (a) Distorted features are utilized for image reconstruction directly. (b) Distorted features have a pre-correction by a predicted appearance flow before reconstructing the corrected image.

*Equal Contributions;
†Corresponding author: cylin@bjtu.edu.cn

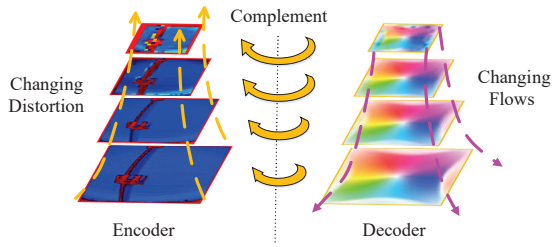


Figure 2. **Gradual generation characteristic.** From low-level to high-level, the distortion of encoder features shows a gradual slight reduction, as well as the displacement of decoder flows.

which is termed as the distortion diffusion problem.

In this paper, we propose a feature-level distortion rectification network[1], which separates the structure correction and content reconstruction, as shown in Fig. 1b. It contains two modules: flow estimation and distortion correction. First, the flow estimation module estimates the distorted image structure and represents the result with dense appearance flows. Second, the distortion correction module leverages flows to correct the distorted features and uses the corrected features to reconstruct a plausible result. To bridge the two modules, we introduce a progressively complementary mechanism to achieve multi-level correction, as shown in Fig. 2. According to our observation, from low-level to high-level, the encoder features on the distortion correction module show a gradual slight reduction in distortion. In the meantime, the decoder outputs at the flow estimation module also decrease progressively on displacement, which is termed as gradual generation characteristic. Therefore, the flow of each layer at the decoder can be used to correct corresponding feature maps, thus solving the problem of distortion diffusion and enhancing the performance. In addition, we propose a multi-scale loss for better supervision of corrected features. Experimental results show that our proposed method obtains superior performance, compared with state-of-the-art methods.

We summarize our contribution as follows:

- Feature-level distortion rectification scheme is proposed for the first time. Feature correction layers are embedded in skip-connection for feature precorrection, which helps the decoder to reconstruct a plausible result.
- The proposed unsupervised flow estimation module is able to estimate the distorted image structure. It can be trained in an end-to-end manner like a self-attention module [21].
- Taking advantage of the gradual generation characteristic, the correction in our network is progressive and complementary. Moreover, a multi-scale loss is introduced to supervise the corrected features.

[1]Available at https://github.com/uof1745-cmd/PCN

2. Related Work

Distortion rectification plays an effective role to bridge the fisheye images and computer vision technology. Traditional methods [22][23][24][25][26] can complete calibration by finding the corresponding feature points from different perspectives. However, such methods required special chessboards and human intervention. Therefore, automatic correction methods [27][28][12][13][29][30] had been made to solve these problems. Depending on the principle that straight lines have to be straight [31], they leveraged special detection methods to detect characteristic curves and then obtained the distortion parameters by calculating the curvature of curves. However, it was vulnerable due to the unstable number of characteristics. Deep learning methods [32][33][34][35][36][37] solved the severe problems that remain in traditional automatic correction methods. Particularly, according to different networks, we categorized deep learning methods into two types, regression-based methods and generation-based methods.

Regression-Based Methods. Regression-based methods utilized a convolutional neural network(CNN) [38] to predict complex nonlinear model parameters. Rong et al. [14] pioneered to train the network on fitted data and used AlexNet to correct the distorted images. However, the limited discrete interval of parameters caused the trained network to perform poorly on complex fisheye images. Yin et al. [16] proposed a multicontext collaborative network, but semantic features can only provide limited guidance because of high dimensional features. Xue et al. [17] imposed explicit geometry constraints to improve the network perception of distorted images. Although achieving better performance, it required a vast amount of labels, such as edge labels, distortion parameter labels, and normal images. Besides, the edge estimation network needed to be pre-trained, which brings a more complex operation.

Generation-Based Methods. The corrected image was directly generated with the help of a generative adversarial network(GAN) [39]. DR-GAN [18] was the first adversarial framework for radial distortion rectification. It can directly learn the distribution pattern between distorted images and normal images instead of estimating the parameters. It achieved label-free training and one-stage rectification. However, the network was overburdened for rebuilding image content and structure simultaneously. The image content was blurred and the structure cannot be completely corrected. Liao et al. [19] proposed a model-free distortion rectification framework for the single-shot case, bridged by the distortion distribution map. It yielded a more accurate correction on the distorted structure. However, cascade network [19] caused image details lost easily, and general skip-connection led to distortion diffusion.

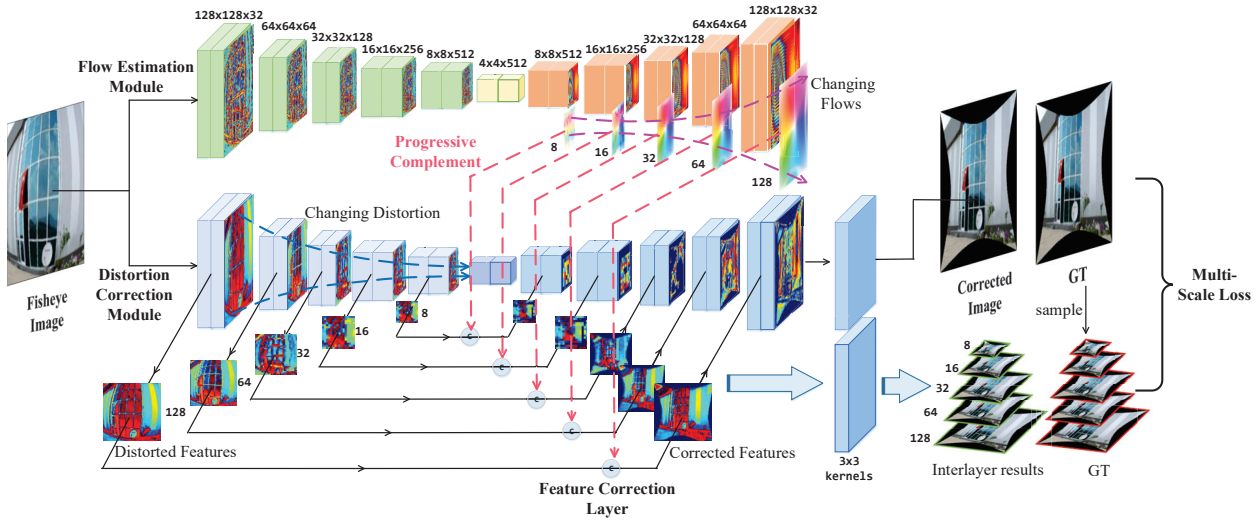


Figure 3. Overview of our complementary network. The network architecture is composed of a flow estimation module (top) and a distortion correction module (bottom). The flow estimation module estimates the structure of the distorted image and provides a series of flows on each decoder layer. The distortion correction module leverages flows to correct corresponding distorted features in the correction layer. The corrected image features are supervised by a multi-scale loss to enhance the performance.

3. Fisheye Models for Synthetic Data

Obtaining massive real distorted images and their corresponding labels are labor-intensive. Therefore, generating synthetic distorted images for training with the fisheye camera model [14][16][17][19] has become a mainstream approach. Generally, division model [40] and polynomial model [41] are the most popular types. In image coordinate system, the Euclidean distance between an arbitrary point $P_u\,(x, y)$ and image center $P_0\,(x_0, y_0)$ on perspective image can be represented as r_u. P_u has a corresponding point $P_d\,(x_d, y_d)$ in fisheye image. Similarly, the Euclidean distance between P_d and distortion center is labeled as r_d. The mapping relationship between r_u and r_d can be represented by division model [40] as follows

$$r_u = \frac{r_d}{1 + \sum_{i=1}^{n} k_i r_d^{2i-1}} \tag{1}$$

Where k_i is distortion parameter. The distortion degree of fisheye image can be variable by changing the value of k_i. n is the number of parameters. Generally, the larger the n is, the more complex distortion state could be represented by polynomial. Compared with division model, polynomial model [41] is more special by involving the angle of incident light. The polynomial model is usually expressed as follows

$$\theta_u = \sum_{i=1}^{n} k_i \theta_d^{2i-1}, \qquad n = 1, 2, 3, 4, \ldots \tag{2}$$

θ_u represents the angle of incident light and θ_d is the angle that light pass through the lens. Generally, r_d and θ_d satisfy the equidistant projection relation, in which $r_d = f\theta_d$. f is the focal length of fisheye camera. As for the pinhole camera, the projection model corresponds to $r_u = f \tan\theta_u$. We simplify the formula and get $\theta_u = \arctan\left(\frac{r_u}{f}\right) \approx \frac{r_u}{f}$. Therefore, we can calculate the relationship between r_u and r_d on polynomial model

$$r_u = f \sum_{i=1}^{n} k_i r_d^{2i-1}, \qquad n = 1, 2, 3, 4, \ldots \tag{3}$$

We merge the k_i and f to get the final polynomial model

$$r_u = \sum_{i=1}^{n} k_i r_d^{2i-1}, \qquad n = 1, 2, 3, 4, \ldots \tag{4}$$

In this paper, the polynomial model is selected to generate the synthesized fisheye images.

4. Proposed Approach

4.1. Network Architecture

Existing generation-based methods simultaneously reconstruct the structure and content of the image on the decoder, which leads to overburden for the decoder and causes blurred results. We separate structure correction and content reconstruction into two modules. As shown in Fig. 3, our network consists of two main components, appearance flow estimation module and distortion correction module. Given a fisheye image with a size of 256×256, we fed it into two modules simultaneously. The appearance flow estimation

module evaluates the distortion degree and presents it as appearance flows. The distortion correction module extracts features on the encoder. Since encoder features contain distortion, we treat them as distorted features. Each layer features are sent to the feature correction layer, leveraging the corresponding appearance flows to pre-correct. Thereafter, with the help of the corrected features, the decoder can concentrate on content reconstruction.

Appearance Flow Estimation Module. Benefiting from the encoder-decoder structure, this module utilizes it to extract features and generate a series of appearance flows. Specially, the output of each decoder layer is involved according to the progressively complementary mechanism that will be detailed later. The output features are experienced additional convolution with 3×3 kernels to obtain two-channel appearance flows. In this way, we obtain 5 appearance flows with sizes of 128, 64, 32, 16, 8. The process can be expressed as

$$\left\{A_f^i\right\}_{i=1}^5 = G_s\left(I_{in}\right) \tag{5}$$

Where I_{in} is the input fisheye image. G_s presents the appearance flow estimation module. A_f^i is the output appearance flow of the i-th decoder layer.

Feature Correction Layer. To solve the distortion diffusion problem, we insert feature correction layer in skip connection, intending to pre-correct the image features before delivering. The predicted flow A_f^i is leveraged to perform spatial transformation [21] as follow

$$I_c^i\left(u,v\right) = I_f^i\left(u + A_f^i\left(u\right), v + A_f^i\left(v\right)\right) \tag{6}$$

I_f^i is the i-th layer feature map in distortion correction module, and the corrected feature map is I_c^i.

Progressively Complementary Mechanism. Based on the visualizing of the feature maps, we impose a progressively complementary mechanism. As shown in Fig. 4, at the encoder of distortion correction module, continuous convolution and pooling operations blur the feature maps edges, making the degree of distortion appear to be slightly reduced. At the decoder of the flow estimation module, the distorted features transferred by skip-connection forces the predicted flows to have a greater displacement to represent greater distortion. Intuitively, we can represent the degree of distortion and displacement as follows

$$\begin{aligned} c \geq D\left(I_f^1\right) \geq D\left(I_f^2\right) \geq D\left(I_f^3\right) \\ \geq D\left(I_f^4\right) \geq D\left(I_f^5\right) \geq 0 \end{aligned} \tag{7}$$

$$\begin{aligned} k \cdot c \geq M\left(A_f^1\right) \geq M\left(A_f^2\right) \geq M\left(A_f^3\right) \\ \geq M\left(A_f^4\right) \geq M\left(A_f^5\right) \geq 0 \end{aligned} \tag{8}$$

Where D is a function of that estimate the degree of input feature distortion. M is a function of estimating the displacement degree of appearance flow. The bigger the distortion of an image, the greater the displacement is required

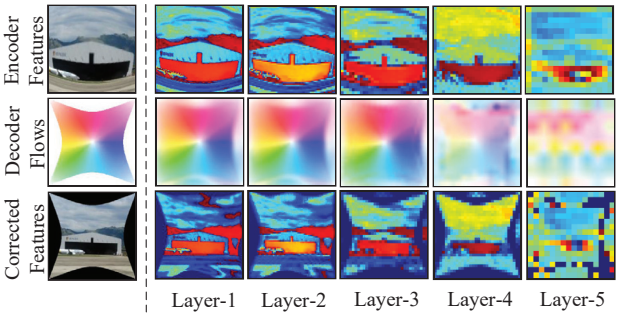


Figure 4. **Progressively complementary mechanism.** The progressive changing flows are leveraged to correct the progressive changing distortion features.

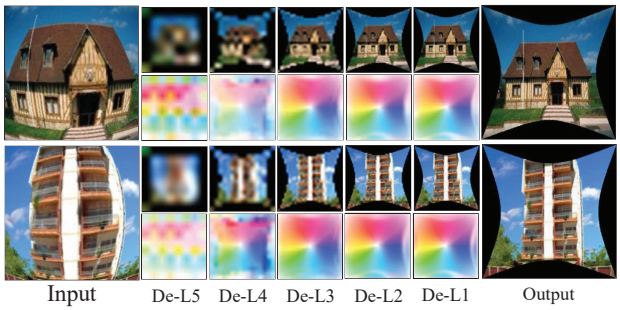


Figure 5. **Multi-scale corrected images.** With the help of progressively complementary mechanism, the feature map on each decoder layer is pre-corrected.

for correction. Moreover, c and k are constant and we have $M\left(A_f^i\right)_{max} / D\left(I_f^i\right)_{max} = k$.

As seen from the formula, there is a significant hierarchical correspondence between I_f^i and A_f^i. They share the same size with 128, 64, 32, 16, 8. Besides, powerful learning ability of the network can build a relationship with $D\left(I_f^i\right) \propto M\left(A_f^i\right)$, which is progressive complement. Thereafter, we can leverages A_f^i in feature correction layer to correct I_f^i.

Distortion Correction Module. With the help of the progressively complementary mechanism, the feature maps used for concatenation on the decoder have been roughly corrected. The corrected feature map can bring a lot of details, without transmitting the distortion structure. Therefore, the network can generate a more visually realistic corrected image. The process of this module can be denoted as

$$\left\{I_{out}^i\right\}_{k=1}^6 = G_c\left(I_{in}, \left\{A_f^i\right\}_{k=1}^5\right) \tag{9}$$

Where I_{out}^i denotes the i-th layer corrected image. G_c indicates the distortion correction module. To ensure the correction quality, we send the concatenated features to the convolutional layer with 3×3 kernels to obtain multi-scale corrected images and downsample the ground truth image at the same scales to supervise them. The multi-scale corrected images can be shown in Fig. 5. As a result, the integrated

network can achieve better end-to-end training.

4.2. Training strategy

Our progressively complementary network is obtained by paralleling two subnetworks. Considering the complexity in the complementary network, we propose a progressively complementary training strategy that contains multiple loss functions to achieve stable end-to-end training.

Reconstruction Loss. Generally, the corrected image not only needs to have a normal structure but also needs to have better image details. Therefore, we first utilize the reconstruction loss to ensure the structural similarity between the corrected image and ground true image. We denote the corrected image as I_{out}. The ground true corrected image is I_{gt}. Therefore, reconstruction loss can be formulated as

$$\mathcal{L}_r = \|I_{out} - I_{gt}\|_1 \tag{10}$$

Adversarial Loss. Reconstruction Loss greatly helps generate the structure of the image, but it is powerless in generating the texture details. Subsequently, we use adversarial loss to enhance the image texture. Adversarial loss can be represented as follow

$$\begin{aligned}\mathcal{L}_{adv} = \min_{G_c}\max_{D} \left(E\left[logD\left(I_{gt}\right)\right] + \right. \\ \left. E\left[log\left(1 - D\left(G_c\left(I_{in}\right)\right)\right)\right]\right)\end{aligned} \tag{11}$$

Where G_c and D denotes the generator and discriminator in distortion correction module respectively.

Enhanced Loss. We introduce an enhanced loss to further enrich texture details. Specifically, enhanced loss consists of content loss and style loss[42]. Content Loss can be denoted as

$$\mathcal{L}_c^j = \frac{1}{C_j H_j W_j} \left\|\phi_j\left(I_{out}\right) - \phi_j\left(I_{gt}\right)\right\|_2^2 \tag{12}$$

$\phi_j(x)$ is the j-th layer feature map of the pre-trained VGG-16 network. It has the shape of (C_j, H_j, W_j). Style Loss can be calculated as follow

$$\mathcal{L}_s^j = \left\|G_j^{\phi}(I_{out}) - G_j^{\phi}(I_{gt}))\right\|_F^2 \tag{13}$$

It is the squared frobenius norm of two gram matrices $G_j^{\phi}(x)$. $G_j^{\phi}(x)$ is a matrix with the shape of (C_j, C_j), and its elements are

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \tag{14}$$

Therefore, our enhanced loss can be recorded as

$$\mathcal{L}_e = \mathcal{L}_c + \lambda_s \mathcal{L}_s \tag{15}$$

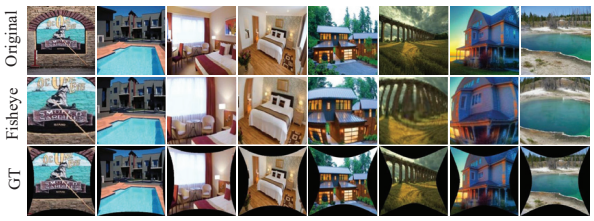


Figure 6. **Synthetic fisheye dataset.** Top: Original Place2 dataset [43]. Middle: Generated fisheye images with different distortion. Bottom: Ground truth corresponding to the fisheye images.

Where λ_s are hyper-parameters, which will be discussed later.

Multi-scale Loss. Our correction is on feature-level. Ideally, the distortion of each layer features should be minimized. Therefore, we propose a multi-scale loss to further enhance the quality of the corrected feature maps. Particularly, multi-scale loss can be expressed by the following formula

$$\mathcal{L}_m = \sum_{i=1}^{L-1} \left\|S\left(I_{gt}, i\right) - C\left(I_c^i \oplus I_d^i\right)\right\|_1 \tag{16}$$

L is the number of convolution blocks. Specifically, we set L to 6. S is the downsampling operation. $S\left(x, n\right)$ represents downsampling the input x to the original $1/2^n$ times. I_d^i and I_c^i represent the original features at the decoder and the features corrected by feature correction layer, respectively. $\oplus$ denotes feature concatenation. C is 3×3 convolution for decoding the output features into 3-channel RGB images. In this way, each feature map on the decoder can be effectively supervised.

Overall Loss Function. The loss function for training the complementary network is obtained by combining the reconstruction loss, adversarial loss, enhanced loss and multi-scale loss. We define the overall loss function as

$$\mathcal{L} = \lambda_r \mathcal{L}_r + \mathcal{L}_{adv} + \lambda_m \mathcal{L}_m + \mathcal{L}_e \tag{17}$$

λ_r, λ_m are hyper-parameters for balancing different loss functions. Through the overall loss function, we can achieve joint training of complementary networks.

5. Experiments

5.1. Dataset and Implementation details

The fisheye dataset with corresponding labels is scarce. Therefore, we use a fisheye model to generate a synthetic fisheye dataset and select the Place2 dataset [43] as our original data, as shown in Fig. 6. The Place2 dataset covers 400 types of scenes and contains 10 million images. We randomly select 100K pictures for training, 8k pictures for testing, and resize the generated fisheye image to 256×256. Like most existing rectification methods

Figure 7. **Comparison on synthetic images.** Visual results comparision in different scenarios. The state-of-the-art methods include: two traditional methods(Auto-DE [11], Auto-DC [44]), two regression-based methods(DC-CNN [14], DeepCalib [15]), three generation-based methods(DR-GAN [18], Blind [20], DDM [19]).

Table 1. Comparison between the proposed method and the state-of-the-art methods on different image complexity.

Methods \ Metrics	$N \leq 200$ (30%)				$200 \leq N \leq 400$ (40%)				$N \geq 400$ (30%)			
	PSNR	SSIM	FID	CW-SSIM	PSNR	SSIM	FID	CW-SSIM	PSNR	SSIM	FID	CW-SSIM
Auto-DE [11]	9.16	0.1964	301.9	0.4746	8.82	0.1478	328.7	0.4611	8.79	0.1073	366.5	0.4673
Auto-DC [44]	9.27	0.2005	298.5	0.4771	8.95	0.1538	325.9	0.4612	8.91	0.1129	361.6	0.4712
DC-CNN [14]	13.83	0.4111	97.8	0.6375	13.42	0.3704	93.6	0.6249	13.01	0.3067	97.0	0.6128
Deep-Calib [15]	20.59	0.6724	69.7	0.8383	18.34	0.5802	82.1	0.8063	18.83	0.5464	69.1	0.8129
DR-GAN [18]	17.23	0.5316	79.6	0.7794	16.57	0.5120	82.3	0.7528	16.24	0.5012	82.5	0.7387
Blind [20]	20.00	0.7047	124.6	0.8343	19.01	0.6317	126.9	0.8153	18.62	0.6004	128.9	0.8139
DDM [19]	22.16	0.7538	55.4	0.9313	21.61	0.7339	59.3	0.9148	20.64	0.6875	59.9	0.9052
Ours	**25.06**	**0.8732**	**25.1**	**0.9615**	**24.99**	**0.8747**	**25.9**	**0.9648**	**24.87**	**0.8770**	**30.0**	**0.9657**

Figure 8. Additional comparison between DDM [19] and our method. Our results provide more details.

[16] [18] [17], our polynomial model contains 4 parameters $P_d = [k_1, k_2, k_3, k_4]$. The value of P_d selection is followed [18] [19] to obtain fisheye images with different distortion degree. We empirically set the hyper-parameters λ_r, λ_m, λ_s to 60, 5, 2500 in overall loss function, respectively. In addition, we use the Adam algorithm with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to optimize our complementary network and set the initial learning rate to 10^{-4}. NVIDIA GeForce GTX 2080Ti GPU is leveraged to train our network.

5.2. Experimental Evaluation

Evaluation Matrics. Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are the two most popular evaluation metrics for images. PSNR can effectively measure the detailed quality of the image, and SSIM can intuitively assess the image structure. In addition to PSNR and SSIM, Fréchet Inception Distance (FID) [45] can measure the difference between two distributions with the help of Wasserstein-2 distance, while complex wavelet structural similarity (CW-SSIM) [46] can evaluate the quality on different geometric transformation. Therefore, we leverage them to objectively evaluate our experimental results.

Comparing Methods. To evaluate the performance, our method are compared with several state-of-the-art methods including: traditional methods(Auto-DE [11], Auto-DC [44]), regression-based methods(Deep-Calib [15], DC-CNN [14]) and generation-based methods(Blind [20], DR-GAN [18], DDM [19]).

Quantitative Comparison Results. Our quantitative comparison results are shown in Tab. 1. To verify the robustness of the method, we leverage the harris algorithm to detect the interest points and divide the test dataset into 3 parts according to the number of corners detected. N is the number of corners. The size of N represents the complexity of the image scene. As reported in Tab. 1, the traditional methods (Auto-DE [11], Auto-DC [44]) obtain poor perfor-

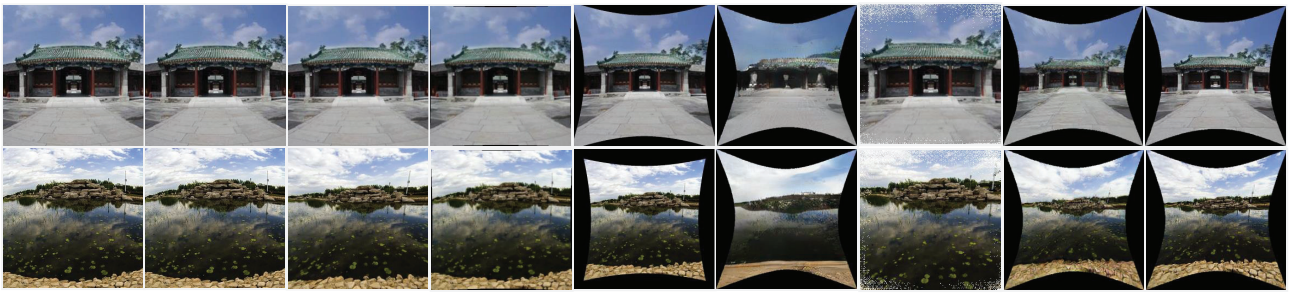


Figure 9. **Comparison on real fisheye images.** Our method outperforms the state-of-the-art methods both in corrected structure and image quality.

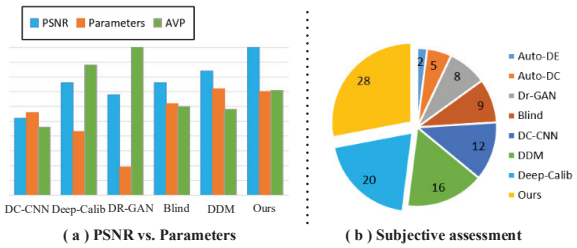


Figure 10. **Network comparison and result evaluation.** (a) A comparison of the PSNR, the number of parameters and AVP (average parameter performance) between deep learning correction methods. (b) Subjective assessment for the results.

Figure 11. A comparison of results based on different architecture.

mance. Regression-based methods(DC-CNN [14], Deep-Calib [15]) are better than the traditional methods. Deep-Calib [15] also transcends the generation-based methods (Dr-GAN [18], Blind [20]) except for DDM [19]. Nevertheless, our method outperforms the above methods. In all the cases of $N \leq 200$, $200 \leq N \leq 400$, and $N \geq 400$, our method achieves the best performance, which fully proves the superiority of our method.

Qualitative Comparison Results. For a more intuitive comparison of corrected results, we visualize the results on the synthetic dataset (Fig. 7). The correction effect of Auto-DE [11] and Auto-DC [44] are not obvious for rarely detected features. Blind [20] is difficult to correct images with large distortion. Deep-Calib [15] well corrects the center region of the image while degrades in the boundary regions. DC-CNN[14] does not fail in boundary regions, but the correction is not complete. DR-GAN[18] is limited by the characteristics of the network itself, the generated image exhibits blur. DDM[19] improves quality by superimposing different features to instruct the decoder. Although the structure correction of our method is similar to DDM[19], the corrected content of our method provides more details. As shown in Fig. 8, local regions of the generated images in DDM[19] have poor quality, while our results provide richer texture information.

Comparison on real fisheye images. Additional experimentation on real fisheye images is necessary. To further verify the practicality of the method, we leverage the synthetic data to train the network and then test the model on a real dataset. Compared with the state-of-the-art methods in Fig. 9, the corrected results of our method are subjectively better. Although there is a gap between the domain of synthetic dataset and real dataset, our method minimizes the gap by separating the content reconstruction from the structure correction and thereby achieves better performance.

Average parameter performance. We propose an average parameter performance (AVP), a novel evaluation metric, to reflect the performance improvement brought by each parameter. With the help of AVP, we can compare the efficiency of deep learning networks. AVP can be calculated by $PSNR/S$, where S is the size of the network model. As shown in Fig. 10a, our PSNR is in the best position and our AVP is in the third position, which proves that our network is effective and efficient.

Subjective assessment. We discover that some methods with lower quantitative performance have plausible visual results, like Deep-Calib [15]. Therefore, to further evaluate the results, we conduct a subjective assessment. We assigned 30 volunteers for the subjective assessment. They are required to select the best rectification results among all comparison methods from 100 fisheye image, which are randomly selected. Then, we averaged the votes of all volunteers and obtained the final subjective evaluation. As shown in Fig. 10b, although Deep-Calib [15] outperforms other methods, the subjective assessment of our corrected

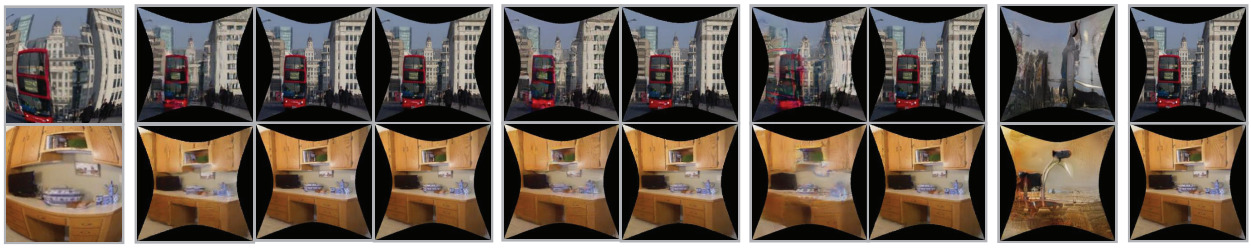


Figure 12. A comparison results of without correcting image features on different layers.

Table 2. Performance comparison for different structures and loss functions.

Methods	PSNR	SSIM	FID	CW-SSIM
w/o FEM	16.09	0.4871	191.8	0.7431
w/o DCM	19.12	0.6605	159.6	0.8784
w/o FCL	21.31	0.7009	86.4	0.9187
w/o MS&EH	23.26	0.7872	36.2	0.9380
w/o EH	23.69	0.7917	32.3	0.9400
Ours	**24.98**	**0.8750**	**26.9**	**0.9640**

results is still the best and exceeds Deep-Calib [15] method by 8%.

5.3. Ablation Study

To verify the effectiveness of each module, we decompose the network and then demonstrate its contribution.

Structure And Loss Ablation. The distortion correction module can independently correct the fisheye image. Therefore, we first remove the module, intending to verify the necessity of the flow estimation module (w/o FEM). Second, we remove the distortion correction module (w/o DCM) and feature correction layer (w/o FCL) respectively. The results are shown in Tab. 2 and Fig. 11. Simply superimposing the appearance flow improves the performance compared with the independent distortion correction module. However, with the help of the feature correction layer, the network can make full use of appearance flow to achieve more accurate correction. Besides, the results in Tab. 2 prove that our loss functions multi-sale loss (MS) and enhanced loss (EH) can further improve network performance.

Flow Ablation. Our complementary network corrects the image features of all encoder layers, but the contribution of correcting layers is still ambiguous. Therefore, we analyze the impact as shown in Tab. 3 and Fig. 12. The results demonstrate the network has the worst performance without correcting all layers (w/o layers all). Correcting without the outermost layers (w/o layer 1) is worse than that correcting without the innermost layers (w/o layer 5). The reason is that compared to the inner image features (layer 5), the outer image features (layer 1) have greater distortion and more detailed information. Therefore, the outer image features (layer 1) need to be corrected more urgently than the inner image features (layer 5). In addition, we observe that the performance of correcting without the innermost layers (w/o layer 5) is similar to that of correcting all network layers (Ours). It further proves that the inner image features (layer 5) have slight distortion. Nonetheless, correcting them can also bring certain performance improvements.

Table 3. Performance of using appearance flow on different convolutional layers.

Methods	PSNR	SSIM	FID	CW-SSIM
w/o Layer(1)	21.23	0.6878	90.3	0.9157
w/o Layer(3)	22.95	0.7835	36.4	0.9365
w/o Layer(5)	23.61	0.8064	33.9	0.9486
w/o Layers(1,2)	21.20	0.6831	87.5	0.9164
w/o Layers(4,5)	23.16	0.8031	33.6	0.9426
w/o Layers(1,2,3)	19.64	0.6048	132.7	0.8823
w/o Layers(3,4,5)	22.63	0.7812	40.1	0.9362
w/o Layers(all)	16.09	0.4871	191.8	0.7431
Ours	**24.98**	**0.8750**	**26.9**	**0.9640**

6. Conclusion

In this paper, we propose a progressively complementary network to correct fisheye images. Two modules are connected in parallel, which can correct the fisheye image and estimate the distortion structure simultaneously. Different from the existing generation-based methods, we uniquely insert the feature correction layer into the skip connection in our network. Pre-correction is implemented before transferring the features, which fundamentally solves the problem of distortion diffusion and implements a feature-level correction. Particularly, taking advantage of the progressive generation characteristics, we design two modules as a novel complementary structure and introduce a multi-scale loss function to supervise the corrected image features. It further enhances the quality of the corrected image. The experimental results of our network significantly outperform the state-of-the-art methods, both subjectively and objectively.

Acknowledgments: This work was supported in part by Fundamental Research Funds for the Central Universities (2020YJS028), in part by National Natural Science Foundation of China (No.61772066, No.62072026) and Beijing Natural Science Foundation (JQ20022).

References

[1] M. Khan, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. Baik. Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:1419–1434, 2019. 1

[2] G. Andreas, L. Philip, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *CVPR*, pages 3354–3361, 2012. 1

[3] S. E. Mahmoodi, R. N. Uma, and K. P. Subbalakshmi. Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Transactions on Cloud Computing*, 7:301–313, 2019. 1

[4] S. Zhang, H. Yao, X. Sun, and X. Lu. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46:1772–1788, 2013. 1

[5] X. Li, C. Ma, B. Wu, Z. He, and M. Yang. Target-aware deep tracking. *CVPR*, pages 1369–1378, 2019. 1

[6] 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. 1

[7] W. Bao, W. Lai, X. Zhang, Z. Gao, and M. Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[8] J. Manuel Álvarez, T. Gevers, Y. LeCun, and A. M. López. Road scene segmentation from a single image. In *ECCV*, 2012. 1

[9] J. Fu, J. Liu, T. Haijie, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *CVPR*, pages 3141–3149, 2019. 1

[10] R. Melo, M. Antunes, J. Pedro Barreto, G. Falcão Paiva Fernandes, and N. Gonalves. Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach. *ICCV*, pages 537–544, 2013. 1

[11] F. Bukhari and M. N. Dailey. Automatic radial distortion estimation from a single image. *Journal of Mathematical Imaging & Vision*, 45(1):31–45, 2013. 1, 6, 7

[12] 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. 1, 2

[13] 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. 1, 2

[14] J. Rong, S. Huang, Z. Shang, and X. Ying. Radial lens distortion correction using convolutional neural networks trained with synthesized images. In *ACCV*, 2016. 1, 2, 3, 6, 7

[15] 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. 1, 6, 7, 8

[16] 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. 1, 2, 3, 6

[17] Z. Xue, N., G. Xia, and W. Shen. Learning to calibrate straight lines for fisheye image rectification. *CVPR*, pages 1643–1651, 2019. 1, 2, 3, 6

[18] 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. 1, 2, 6, 7

[19] 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. 1, 2, 3, 6, 7

[20] 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. 1, 6, 7

[21] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 2, 4

[22] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 1:666–673 vol.1, 1999. 2

[23] 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. 2

[24] S. Gasparini, P. F. Sturm, and J. Pedro Barreto. Plane-based calibration of central catadioptric cameras. *ICCV*, pages 1195–1202, 2009. 2

[25] 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. 2

[26] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1330–1334, 2000. 2

[27] 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. 2

[28] M. Stuiver and P. Reimer. Extended 14c data base and revised calib 3.0 14c age calibration program. *Radiocarbon*, 35:215–230, 1993. 2

[29] 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. 2

[30] 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. 2

[31] 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. 2

[32] C. Guo, P. Geoff, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *ArXiv*, abs/1706.04599, 2017. 2

[33] W. Zhang, H. Ren, C. Pan, M. Chen, R. C. D. Lamare, B. Du, and J. Dai. Large-scale antenna systems with ul/dl hardware mismatch: Achievable rates analysis and calibration. *IEEE Transactions on Communications*, 63:1216–1229, 2015. 2

[34] M. Lee, K. Hyungtae, and P. Joonki. Correction of barrel distortion in fisheye lens images using image-based estimation of distortion parameters. *IEEE Access*, 7:45723–45733, 2019. 2

[35] M. Yang, C. Yang, and T. Meng. Correct a particular fisheye lens distortion quickly using the coordinate map table. In *International Conference on Image Processing and Pattern Recognition in Industrial Engineering*, 2010. 2

[36] G. Timothy and N. Grigorieff. Automatic estimation and correction of anisotropic magnification distortion in electron microscopes. *Journal of structural biology*, 192 2:204–8, 2015. 2

[37] C. Ophus, J. Ciston, and C. T. Nelson. Correcting nonlinear drift distortion of scanning probe and scanning transmission electron microscopies from image pairs with orthogonal scan directions. *Ultramicroscopy*, 162:1–9, 2016. 2

[38] A. Krizhevsky, S. Ilya, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *CACM*, 2017. 2

[39] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Andrew, T. Alykhan, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*, pages 105–114, 2017. 2

[40] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. *CVPR*, 1:I–I, 2001. 3

[41] A. Basu and S. Licardie. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16:433–441, 1995. 3

[42] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016. 5

[43] 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. 5

[44] M. Alemánflores, L. Álvarez, L. Gómez, and D.SantanaCedrés. Automatic lens distortion correction using one-parameter division models. *IPOL*, 4:327–343, 2014. 6, 7

[45] 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. 6

[46] 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(11):2385–2401, 2009. 6