

Spatiotemporal Deformation Perception for Fisheye Video Rectification

Shangrong Yang, Chunyu Lin*, Kang Liao, 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, yzhao}@bjtu.edu.cn

Abstract

Although the distortion correction of fisheye images has been extensively studied, the correction of fisheye videos is still an elusive challenge. For different frames of the fisheye video, the existing image correction methods ignore the correlation of sequences, resulting in temporal jitter in the corrected video. To solve this problem, we propose a temporal weighting scheme to get a plausible global optical flow, which mitigates the jitter effect by progressively reducing the weight of frames. Subsequently, we observe that the inter-frame optical flow of the video is facilitated to perceive the local spatial deformation of the fisheye video. Therefore, we derive the spatial deformation through the flows of fisheye and distorted-free videos, thereby enhancing the local accuracy of the predicted result. However, the independent correction for each frame disrupts the temporal correlation. Due to the property of fisheye video, a distorted moving object may be able to find its distorted-free pattern at another moment. To this end, a temporal deformation aggregator is designed to reconstruct the deformation correlation between frames and provide a reliable global feature. Our method achieves an end-to-end correction and demonstrates superiority in correction quality and stability compared with the SOTA correction methods.

Introduction

Many existing algorithms including object detection (Redmon et al. 2016)(Lin et al. 2017) and tracking (Zhang et al. 2013)(Li et al. 2019a), pedestrian re-identification (Pedagadi et al. 2013), and human pose estimation (Newell, Yang, and Deng 2016) are designed based on the perspective camera models. Although they achieve satisfactory performance on ordinary planar video, their field-of-view (FOV) are limited. As a result, in most scenarios, fisheye cameras replace multiple perspective cameras to obtain a wider field-of-view (FOV) at a cheap cost. However, the structure of the fisheye video is different from that of the distorted-free video. If these algorithms are directly applied to fisheye video, the performance will be dramatically degraded. Therefore, it is necessary to pre-correct the video to maintain the performance of subsequent algorithms.

The correction methods for video processing have rarely been investigated, and most of them are based on the image

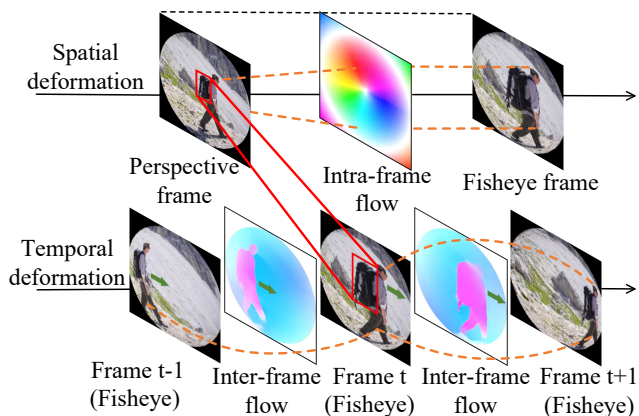


Figure 1: The spatial and temporal deformation. The orange line represents the pixel motion caused by the distortion. The red box indicates that it is possible for the distorted moving object to find its distorted-free appearance at other times.

processing. A series of traditional methods (Reimer et al. 2009)(Zhang 1999) correct fisheye images with the help of calibration boards. They require labor-intensive manual operation. Many self-calibration methods (Dansereau, Pizarro, and Williams 2013)(Melo et al. 2013) can automatically detect significant features to calculate the distortion parameters. However, the detected features are greatly affected by the scene content. With the development of deep learning, many novel networks have been designed for distortion correction. (Rong et al. 2016)(Bogdan et al. 2018)(X.Yin et al. 2018)(Xue et al. 2019) improve the convolutional neural networks (CNNs) to regress the distortion parameters. (Liao et al. 2019)(Liao et al. 2020)(Li et al. 2019b) design the network on the architecture of generative adversarial networks (GANs) to generate rectified images directly. These methods are specially devised for image rectification and do not consider the temporal correlation of fisheye video frames. Different frames generate different results, which cause instability in the corrected video. Achieving consistent correction for each frame is the biggest challenge for existing correction methods.

We regard the pixel motion caused by the distortion of a single image as *intra-frame* optical flow and the motion between different frames as *inter-frame* optical flow. In this

*Corresponding author: cylin@bjtu.edu.cn

paper, we propose a fisheye video correction method based on spatiotemporal deformation perception¹. Considering the output of a stable algorithm should not change significantly due to the input of new frames, we first predict the intra-frame optical flow independently for each frame. Then we use a temporally decreasing weighting scheme to generate a coarse intra-frame optical flow to mitigate the impact of new input. However, independent prediction for intra-frame optical flow ignores the temporal correlation. Since the distortion of the fisheye video increases with the radius, it is possible for the distorted moving object to find its distorted-free appearance at other times, as shown in Figure 1. Therefore, we introduce a temporal deformation aggregator to reconstruct the temporal relationship between the video frames. The reconstructed global feature is conducive to predicting a more accurate intra-frame optical flow. In addition, according to the derivation, we found dual optical flows (the inter-frame optical flow of fisheye video and its corresponding flow of distorted-free video) can effectively reflect object’s spatial deformation. We thereby use the independent intra-frame optical flow to correct the corresponding fisheye frame and estimate dual optical flows. A spatial local deformation attention is then calculated from dual optical flows to enhance the local accuracy of the intra-frame optical flow.

In summary, this work includes the following contributions:

- We pioneer to explore the stable correction on fisheye video and propose a temporal weighting scheme to alleviate the jitter problem in existing correction methods.
- Our network fully perceives the spatiotemporal deformation of objects, which is leveraged to simultaneously enhance the global and local accuracy of the prediction.
- Compared with existing methods, our method fully exploits the spatiotemporal information of the video and achieves promising improvements in both video stability and correction quality.

Related Work

Distortion correction is a meaningful work because it greatly alleviate the impact of distortion on algorithms that are designed for planar cameras, such as object detection (Ren et al. 2015)(Lin et al. 2020), semantic segmentation (Girshick et al. 2014)(Shelhamer, Long, and Darrell 2017) and pedestrian re-identification (Liu et al. 2015)(Han et al. 2021). Most of the existing distortion correction methods (Mei and Rives 2007)(Melo et al. 2013)(Bukhari and Dailey 2013) are designed based for images. At the beginning, distortion correction was simplified to the camera calibration (Zhang 2000)(Luong and Faugeras 2004). However, the method fails on fisheye images for two reasons: (1) Ordinary pinhole camera models cannot express the relationship between fisheye images and 3D scenes; (2) Calculating the parameters of the fisheye camera model (Barreto 2006)(Basu and Licardie 1995)(Alemánflores et al. 2014) requires more detected features. Therefore, Mei et al. (Mei and Rives 2007) proposed a manual calibration method. He leveraged

line image points to calculate the distortion parameters. Melo et al. (Melo et al. 2013) designed an automatic method and greatly improved the smoothness of distortion correction. Nevertheless, in complex scenes, the feature selection has low accuracy, which further affects the performance of correction.

Considering that the neural network can automatically locate the most representative features, many researchers used neural networks to extract features. Rong et al. (Rong et al. 2016), Yin et al. (X.Yin et al. 2018), Xue et al. (Xue et al. 2019) leveraged convolutional neural networks(CNNs) to predict parameters. Although their performance exceeded self-calibration methods (Melo et al. 2013), it was difficult to accurately predict all the parameters. Another researcher used Generative Adversarial Networks (GANs) for distortion correction. They reasoned that the domain gap between the distorted and corrected images is smaller than that between the image and the parameters. Liao et al. (Liao et al. 2019) (Liao et al. 2020), Yang et al. (Yang et al. 2021), Zhao et al. (Zhao et al. 2021) used GANs to implement the transformation from fisheye image to normal image. Compared with the regression method, their correction speed has been largely improved, but the generated content has artifacts.

The aforementioned methods are designed for image correction, while the distortion video correction method is under-explored. Since the video has one more temporal dimension than the image. Therefore, using the image approaches to correct video frame by frame has two problems: (1) Correcting each frame individually, the network produces different outputs due to the varying input content and generates a corrected video with jitter effect. (2) Rectifying the video with uniform transformation, it is difficult to find the best predicted parameter. Therefore, in this paper, we take the temporal dimension into account and design a deep video correction method. Our method solves the jitter problem as well as improves the accuracy of the correction.

Fisheye Model

The camera model reflects the mapping process from a 3D scene to an image. There are three models commonly used to express the mapping relationship of fisheye cameras: the sphere model (Barreto 2006), the polynomial model (Basu and Licardie 1995) and the division model (Alemánflores et al. 2014). The sphere model indicates an excellent fit between the panoramic scene and the fisheye image. However, compared with panoramic images, perspective images are easily accessible. Therefore, many researchers leveraged the division model or the polynomial model to achieve a simpler transformation. Since the polynomial model is similar to the division model and has been frequently used in recent years. Therefore, we leverage the polynomial model to construct our dataset. Generally, the polynomial model can be expressed as

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = (1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + \dots) \begin{bmatrix} x_d \\ y_d \end{bmatrix} \quad (1)$$

Where (x_u, y_u) and (x_d, y_d) are the image coordinates on the perspective image and fisheye image. $[k_1, k_2, k_3, \dots]$

¹Available at <https://github.com/uof1745-cmd/SDP>

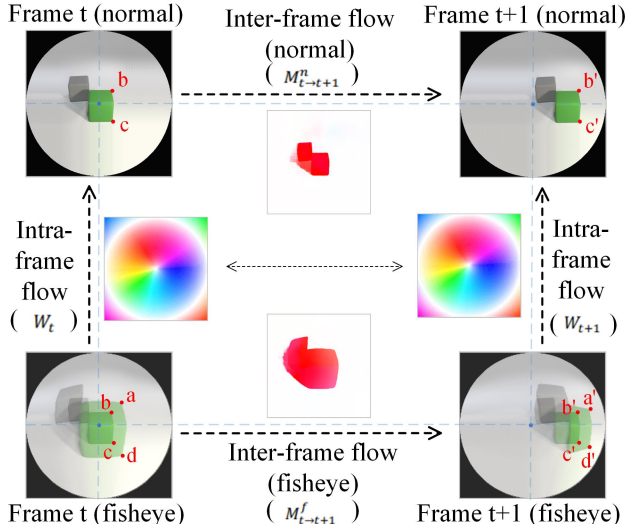


Figure 2: The relationship between the inter-frame flow and the intra-frame flow. The blue dotted lines indicate the alignment of the image center.

are distortion parameters, which can control the distortion degree of the image. The distortion radius is represented by r_d , which is the distance between the distortion coordinates (x_d, y_d) and the distortion center (x_0, y_0) . It can be calculated as

$$r_d = \sqrt{(x_d - x_0)^2 + (y_d - y_0)^2} \quad (2)$$

Dual Optical Flow and Fisheye Deformation

For video sequences, if an object is located in the center of the image in the first frame, it gradually moves towards the boundary in subsequent frames. Since the distortion of fisheye images increases with radius, we can observe that the object gradually deforms. However, even though the object has some displacement, it will have no deformation in the perspective video. By perspective video, we want to denote the video is captured with a perspective camera with no obvious deformation in contrast with fisheye lens. We consider using the displacements of objects in fisheye video and distorted-free video to infer the deformation. As shown in Figure 2, suppose there is a square object at frame t (Johnson et al. 2017) of a perspective video, and its upper right corner and lower right corner are located at pixel points b and c , respectively. If the object has translation, the upper right and lower right corners will move to b' and c' at frame $t+1$. Because the object deforms in fisheye video, distortion points a and d correspond to b and c , respectively. The distortion points of a' and d' correspond to b' and c' at frame $t+1$. We use M^f and M^n to denote the optical flow map of fisheye video and distorted-free video, respectively. $M_{t \rightarrow t+1}^f(x)$ represents the value of M^f at the pixel point x in frame t . $M_{t \rightarrow t+1}^n(x)$ is the pixel point value of M^n in frame t . $L(x)$ is the pixel coordinate of the x , then we have

$$M_{t \rightarrow t+1}^f(a) = \vec{a'a} = L(a') - L(a) \quad (3)$$

$$M_{t \rightarrow t+1}^n(b) = \vec{b'b} = L(b') - L(b) \quad (4)$$

By subtracting these two formulas, we can obtain:

$$M_{t \rightarrow t+1}^f(a) - M_{t \rightarrow t+1}^n(b) = (L(a') - L(b')) - (L(a) - L(b)) \quad (5)$$

We denote the intra-frame optical flow which is the pixel motion caused by the distortion of fisheye video to W . It represents the uniform deformation performed on each fisheye frame. The value of W at the point x in the t frame is $W_t(x)$. $W_t(a')$ and $W_t(a)$ can be specifically expressed as

$$W_t(a') = L(a') - L(b') \quad (6)$$

$$W_t(a) = L(a) - L(b) \quad (7)$$

Then, Equation 5 can be simplified to

$$\Delta M = M_{t \rightarrow t+1}^f(a) - M_{t \rightarrow t+1}^n(b) = W_t(a') - W_t(a) = \Delta W \quad (8)$$

The difference in the intra-frame optical flow can be obtained by subtracting the flow map of the distorted-free video from the corresponding fisheye flow map. The difference between the fisheye and the normal flow map represents the deformation of the object during motion. Therefore, the flow map of fisheye video and preliminary corrected video can benefit the network to perceive the deformation of the object, thereby enhancing the local accuracy of intra-frame optical flow.

Architecture

Compared with image correction, video correction is more challenging because of the additional temporal dimension. To this end, we propose a multi-stage correction network to maintain the stability of corrections in timestamps and achieve smooth transitions between different timestamps. Our multi-stage correction network includes the flow estimation network (FSN) and the flow enhancement network (FEN), as shown in Figure 3. The FSN performs a simple estimation of the intra-frame optical flow for each fisheye frame. The FEN removes the jitter effect with a temporal weighting scheme. In addition, a temporal deformation aggregator (TDA) is presented to enhance global correlation, and a spatial deformation perception module (SDM) is used to enhance local accuracy. Finally, the global optical flow generated by the FEN is used to resample the fisheye video to obtain corrected video frames.

Flow Estimation Network

The purpose of fisheye video correction is to find an intra-frame optical flow for all frame corrections. To achieve this, we first construct a flow estimation network (FSN) to predict the intra-frame optical flow W for each fisheye frame F . The FSN is a pyramid structure that uses 6 convolutional layers to extract features and reconstruct optical flows. The input of the network is 5 consecutive fisheye images $\{F_{t-4}, F_{t-3}, F_{t-2}, F_{t-1}, F_t\}$ with size of 256×256 . The shared-weight FSN makes independent predictions for each frame. The predicted five intra-frame optical flows $\{W_{t-4}, W_{t-3}, W_{t-2}, W_{t-1}, W_t\}$ can be used to describe

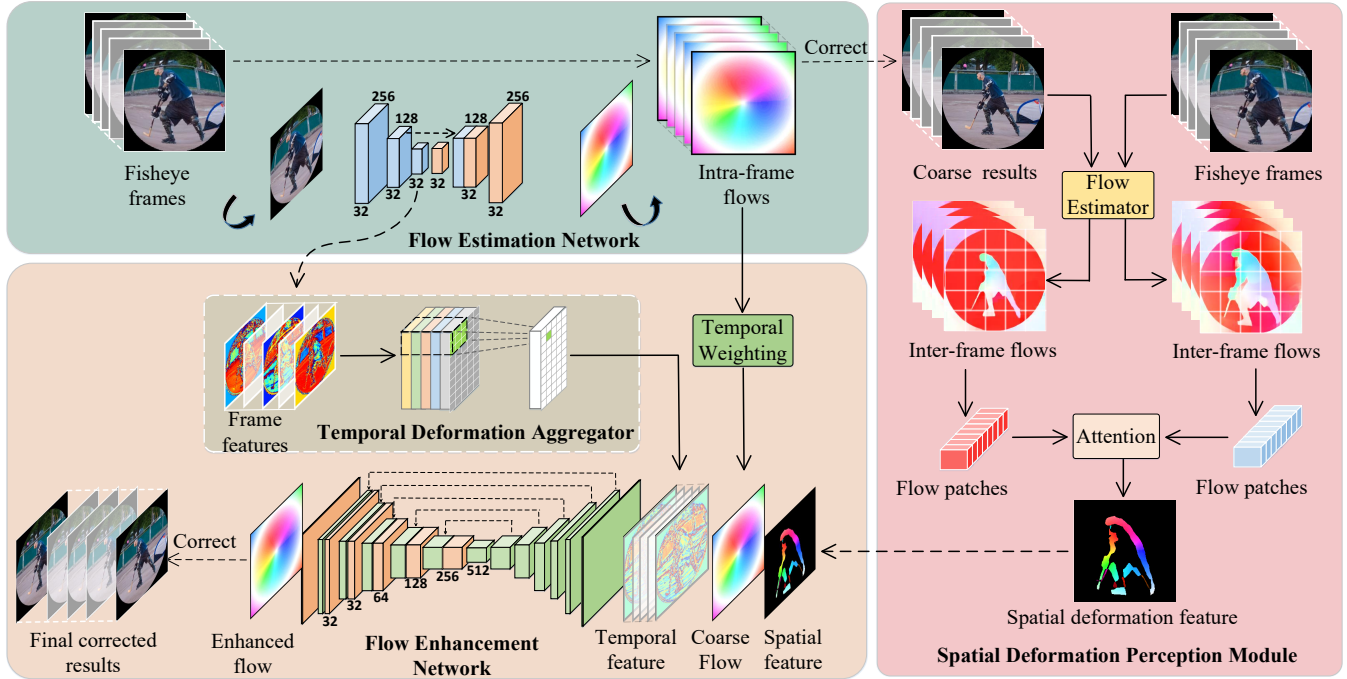


Figure 3: The overview of our network. The flow estimation network predicts intra-frame flows for preliminary correction. A temporal weighting scheme is first used to alleviate the video jitter. Then the extracted features and the preliminary correction results are sent to the temporal deformation aggregator and spatial deformation perception module to establish spatiotemporal correlation. The flow enhancement network benefits from all useful information and generates accurate results.

the deformation of each fisheye frame. Besides, the FSN also outputs the feature map X with a size of $(64, 64, 32)$ on the encoder. Because independent prediction in the FSN disrupts the temporal correlation, we need to establish the temporal correlation for X in the subsequent flow enhancement network (FEN).

Flow Enhancement Network

The predicted intra-frame optical flow varies according to the content of the frame. Then the inconsistency correction on the fisheye frames leads to the video jitter. Therefore, we first design a progressive temporal weighting scheme to perform weighting on $\{W_{t-4}, W_{t-3}, W_{t-2}, W_{t-1}, W_t\}$. The generated unified intra-frame optical flow W' is the input of the flow enhancement network (FEN). Since each W is predicted independently, we introduce a temporal deformation aggregator (TDA) to establish the temporal correlation for the features X . In addition, we observe that the difference between the inter-frame optical flow of the fisheye video $M_{t-1 \rightarrow t}^f$ and preliminary corrected video $M_{t-1 \rightarrow t}^n$ reflects the local deformation of the frame, as derivation in Equation 8. Therefore, we calculate the attention A from $M_{t-1 \rightarrow t}^f$ and $M_{t-1 \rightarrow t}^n$ to enhance the local accuracy of W' . Our FEN has a 6-layer encoder-decoder structure with a kernel size of 3×3 and a stride of 1. Finally, we take W' , temporal aggregation information X' , and spatial deformation feature A as input, and output an improved intra-frame optical flow W^* .

Temporal Weighting Scheme The network outputs different intra-frame optical flows according to different video frames. We need to mitigate the apparent variation between different outputs. Therefore, we design a temporal weighting scheme (TWS). Assuming that we have obtained the intra-frame optical flow of the first fisheye frame W_1 , we treat it as the initial integrated intra-frame optical flow W' . When the second flow W_2 arrives, we need to update W' according to W_1 and W_2 . We use a simple linear weighting method, which can be expressed by

$$W' = aW_1 + (1 - a)W_2, \quad 0 < a < 1 \quad (9)$$

To alleviate the jitter effect, it is necessary to make W' tend to W_1 , which implies $a > 0.5$. The weight of the subsequent flow should be smaller than the previous flow so that the integrated flow does not change greatly due to the new input. We use a progressive temporal weighting scheme, which can be recorded as

$$W' = a_1W_1 + a_2W_2 + \dots + a_{n-1}W_{n-1} + a_nW_n \quad (10)$$

where the weight set $a_1, a_2, \dots, a_{n-1}, a_n$ is an arithmetic progression and their sum is 1. In our experiment, $n = 5$. We empirically set $a_1 = 0.3$. Therefore, for our intra-frame optical flows $\{W_{t-4}, W_{t-3}, W_{t-2}, W_{t-1}, W_t\}$, their corresponding weights are $\{0.3, 0.25, 0.2, 0.15, 0.1\}$.

Temporal Deformation Aggregator The distortion of the fisheye video increases with the radius. The distorted moving object may be able to find its distorted-free appearance at

other times. However, this temporal correlation is broken in the flow estimation network due to independent flow prediction. Therefore, we introduce a temporal deformation aggregator (TDA). With the help of 3D convolution, it performs reconstruction in the temporal dimension for output features of the flow estimation network and obtains a new feature X' with temporal correlation. Finally, we concatenate X' and the weighted intra-frame optical flow W' on the channel and feed them to the flow enhancement network. X' can help the flow enhancement network to improve the global accuracy of the W' .

Spatial Deformation Perception Module According to Equation 8, the difference between the inter-frame optical flow of the fisheye video $M_{t-1 \rightarrow t}^f$ and the preliminary corrected video $M_{t-1 \rightarrow t}^n$ reflects the local deformation. It indicates that we can further enhance the local accuracy of W_f through $M_{t-1 \rightarrow t}^f$ and $M_{t-1 \rightarrow t}^n$. However, to obtain the inter-frame optical flow of the distorted-free video, it is necessary to perform a preliminary correction on the fisheye video. Therefore, in the spatial deformation perception module (SDM), we first use the intra-frame optical flows $\{W_{t-4}, W_{t-3}, W_{t-2}, W_{t-1}, W_t\}$ to correct fisheye frames $\{F_{t-4}, F_{t-3}, F_{t-2}, F_{t-1}, F_t\}$ and obtain the rectified frames $\{I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}, I_t\}$. Subsequently, we leverage a lightweight network Raft (Teed and Deng 2020) to predict the optical flow for fisheye video $M_{t-n \rightarrow t}^f$ and the distorted-free video $M_{t-n \rightarrow t}^n$, respectively. Both $M_{t-n \rightarrow t}^f$ and $M_{t-n \rightarrow t}^n$ have a size with $(n-1, 2, w, h)$, where n, w, h represent the number of frames in a timestamp, the length and width of the video. We use a window with a length of 8 to divide the $M_{t-n \rightarrow t}^f$ and $M_{t-n \rightarrow t}^n$ into patches. The patches have a size of $(n-1 * win, 2, 8, 8)$, where $win = (w/8) * (h/8)$. As the benefit of the fully connected layer, we map the patches of $M_{t-n \rightarrow t}^f$ to the query and value, and the patches of $M_{t-n \rightarrow t}^n$ to the key, to calculate the attention A . In the same way, we concatenate A to the flow enhancement network. With the assistance of attention A , the network can dynamically perceive the deformation and further enhance the local accuracy of the optical flow.

Training Strategy

In the flow estimation network, we use W_t to correct F_t , and obtain a coarse corrected video Y_c . Since we need to estimate the inter-frame optical flow for the corrected video, the supervision of Y_c is necessary. In the flow enhancement network, the outputs W_f is leveraged to correct F_t , resulting in a corrected video Y_r . We first design a reconstruction loss to minimize the L1 distance between Y_c, Y_r and the ground-truth sequence \hat{Y} . It can be expressed as

$$\mathcal{L}_R = \left\| Y_c - \hat{Y} \right\|_1 + \left\| Y_r - \hat{Y} \right\|_1 \quad (11)$$

Our network implements image-to-image transformation using Encoder-Decoder architecture. We thereby use the adversarial loss to align the domains and supervise the final

output Y_r , which can be represented as

$$\mathcal{L}_A = \min_{G_c} \max_D \left(E \left[\log D \left(\hat{Y} \right) \right] + E \left[\log \left(1 - D \left(G_c \left(Y_r \right) \right) \right) \right] \right) \quad (12)$$

In addition, resampling the image with optical flow causes blur, so we enhance the details with style loss \mathcal{L}_S and feature loss \mathcal{L}_F (Johnson, Alahi, and Li 2016) as follow

$$\mathcal{L}_S = \left\| G \left(\phi \left(Y_r \right) \right) - G \left(\phi \left(\hat{Y} \right) \right) \right\|_F^2 \quad (13)$$

$$\mathcal{L}_F = \frac{1}{CHW} \left\| \phi \left(Y_r \right) - \phi \left(\hat{Y} \right) \right\|_2^2 \quad (14)$$

Where $\phi(\cdot)$ is the feature of each layer in the VGG16 network, and $G(\cdot)$ is the feature multiplied by its transpose.

Finally, we combine the abovementioned losses to train our network. The overall loss is:

$$\mathcal{L} = \lambda_r \mathcal{L}_R + \lambda_a \mathcal{L}_A + \lambda_s \mathcal{L}_S + \lambda_f \mathcal{L}_F \quad (15)$$

where $\{\lambda_r, \lambda_a, \lambda_s, \lambda_f\}$ are the hyperparameter. We empirically set $\{20, 0.05, 250, 0.5\}$.

Experiment

Experiment Setup

Since fisheye video correction has rarely been explored, the generation of our dataset is referred to previous fisheye image synthesis methods (Liao et al. 2020)(Yang et al. 2021). We first use DAVIS (Perazzi et al. 2016) to synthesize our fisheye videos. DAVIS includes 90 videos for training and 30 for testing. Additionally, we apply Youtube-VOS (Xu et al. 2018) which contains 3400 training videos and 500 testing videos. We use random parameters to generate 3.5K training timestamps and 0.1K testing timestamps with the video of DAVIS, and generate 50K training timestamps and 3K testing timestamps with the video of Youtube-VOS. In the experiment, we resize the input timestamp to 256×256 and set the batch size to 4. We leverage the Adam optimizer with a learning rate of 1×10^{-4} to train the network on an NVIDIA GeForce RTX 3090 for 30 epochs.

Qualitative and Quantitative Comparison

We perform qualitative and quantitative comparisons with state-of-the-art methods Blind (Li et al. 2019b), DCCNN (Rong et al. 2016), DRGAN (Liao et al. 2019), DeepCalib (Bogdan et al. 2018), DDM (Liao et al. 2020), PCN (Yang et al. 2021). From Figure 4 and Figure 6, DCCNN and DRGAN produce unsatisfactory results, because their simple structure cannot thoroughly learn the transformation from source domain to the target domain. Blind and DeepCalib improves the structure, but they lose the correction boundary. DDM and PCN make full-frame corrections for each pixel preserved, but their discrete edge points and unsupervised flow lack explicit guidance. Therefore, they produce artifacts as shown in Figure 7. In contrast, our method quickly construct the guidance of flow by supervising two transformed video frames. Subsequently, the spatiotemporal deformation perception further enhance the global and local

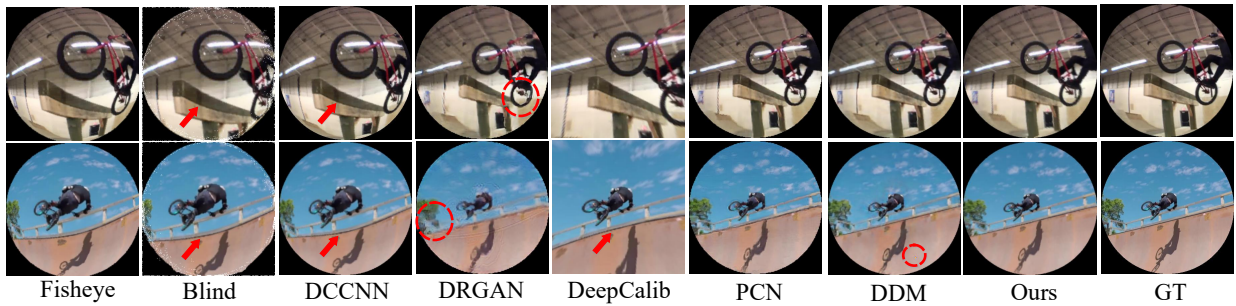


Figure 4: Visualization of qualitative comparison results. The first row demonstrates the correction results on Youtube-VOS video sequences, and the rest are the results on DAVIS video sequences. We marked defect structure and content with red arrows and circles, respectively (zoom in to see the details).

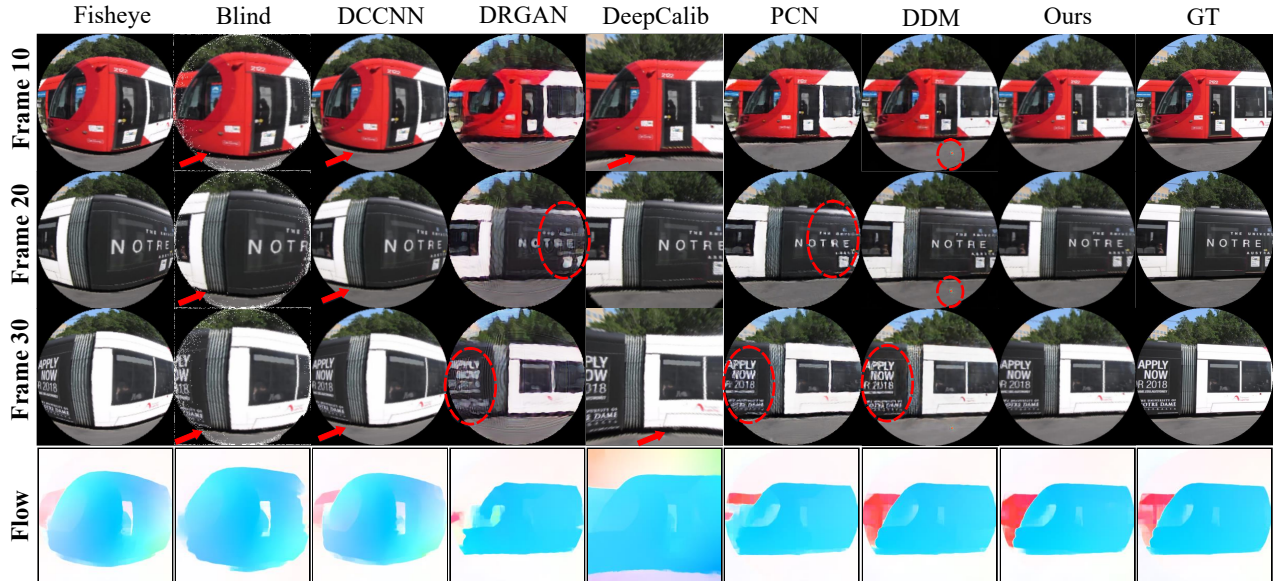


Figure 5: The correction result on the fisheye video. Our correction result maintains stability and local quality. Red arrows and circles highlight the distortion and artifact, respectively. Besides, we visualize part of the optical flows (bottom).

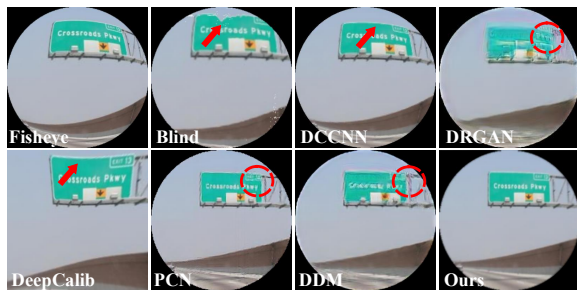


Figure 6: Qualitative comparison on real fisheye videos.

accuracy of flow. Our method thereby achieves better visual performance with accurate structure and realistic texture.

In addition, we also calculate PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity), FID (Fréchet Inception Distance) (Martin et al. 2017), and CW-SSIM (Complex Wavelet Structural Similarity) (Sampat et al. 2009) to objectively evaluate the corrected videos. The comparison of quantitative results in Table 1 intuitively highlights the superior performance of our method.

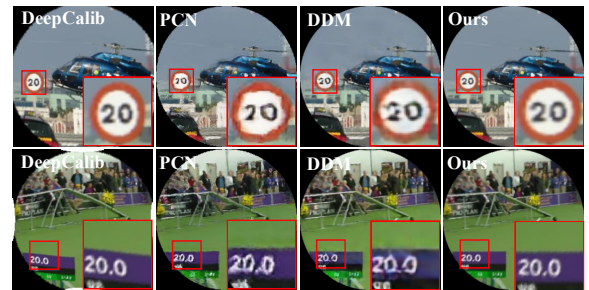


Figure 7: We zoom in the corrected results to demonstrate that our results have better texture details.

Stability Analysis

To verify the stability of our approach, we also test on fish-eye videos. Since the most accurate prediction of intra-frame optical flow is difficult to obtain, we therefore use Blind, DCCNN, DRGAN, DeepCalib, DDM, and PCN to independently predict and correct each frame. The comparison method and our method have one and five input frames, respectively. For a fair comparison, our method also makes

Methods	Dataset	PSNR	SSIM	FID	CW-SSIM	Cropping	Distortion	Stability	EPE
Blind (Li et al. 2019b)	DAVIS	11.33	0.3141	173.5	0.5943	0.807	0.575	0.736	3.97
	Youtube	11.65	0.3164	190.8	0.5934	0.754	0.529	0.683	1.23
DCCNN (Rong et al. 2016)	DAVIS	14.63	0.4231	119.4	0.7222	0.813	0.398	0.733	3.01
	Youtube	14.99	0.4252	125.7	0.7113	0.828	0.421	0.781	1.01
DRGAN (Liao et al. 2019)	DAVIS	17.09	0.4714	194.5	0.8269	0.822	0.673	0.726	2.01
	Youtube	15.67	0.4058	212.9	0.7387	0.780	0.619	0.676	0.88
DeepCalib (Bogdan et al. 2018)	DAVIS	15.96	0.5198	73.7	0.7841	0.911	0.677	0.767	7.46
	Youtube	14.72	0.4781	123.6	0.6393	0.836	0.558	0.679	5.51
PCN (Yang et al. 2021)	DAVIS	22.95	0.7980	86.1	0.9537	1.000	0.916	0.869	1.36
	Youtube	24.98	0.8182	46.2	0.9512	1.000	0.965	0.892	0.35
DDM (Liao et al. 2020)	DAVIS	24.90	0.8317	63.2	0.9614	0.988	0.957	0.877	1.58
	Youtube	26.42	0.7948	47.2	0.9446	0.990	0.959	0.865	0.42
Ours	DAVIS	25.92	0.8731	45.4	0.9727	1.000	0.966	0.888	0.84
	Youtube	28.18	0.8889	37.2	0.9701	1.000	0.974	0.895	0.24

Table 1: Performance comparison with different methods.

	PSNR	SSIM	FID	CW-SSIM	Cropping	Distortion	Stability
w <i>Flow Estimation Network</i>	21.23	0.7334	91.5	0.9001	0.990	0.903	0.865
w <i>Flow Estimation + Enhancement</i>	28.18	0.8889	37.2	0.9701	1.000	0.974	0.895
w/o <i>Temporal Weighting Scheme</i>	26.98	0.8770	45.9	0.9668	1.000	0.974	0.884
w/o <i>Temporal Deformation Aggregator</i>	27.54	0.8826	38.1	0.9675	1.000	0.973	0.892
w/o <i>Spatial Deformation Perception</i>	27.03	0.8771	46.2	0.9663	1.000	0.972	0.891

Table 2: The structure performance of different key component.

independent predictions for each frame. In a video, we generate multiple timestamps with overlapping by assuming the first timestamp is $\{F_{t-4}, F_{t-3}, F_{t-2}, F_{t-1}, F_t\}$ and the next timestamp is $\{F_{t-3}, F_{t-2}, F_{t-1}, F_t, F_{t+1}\}$. Subsequently, we take the last frame of each corrected timestamp as the result of an independent prediction. Figure 5 shows the correction results of some frames in the same fisheye video. It can be seen that DCCNN, DRGAN, Blind, DeepCalib, DDM, and PCN have large differences in the correction of different frames, while our method achieved better stability with the help of a temporal weighting scheme. In order to compare the stability more intuitively, we use common evaluation metrics (cropping, distortion, and stability) (Liu et al. 2013) in video stabilization to measure the stabilization performance of the corrected video. In Table 2, our method is higher than other methods in cropping, distortion, and stability, which further indicates the effectiveness of our method.

Testing on Downstream Tasks

Testing on downstream tasks can verify the quality of the corrected video. Therefore, we use Raft (Teed and Deng 2020) to estimate the optical flow from corrected videos (in Figure 5), and calculate their EPE (end-point-error) (in Table 1). Our method has the lowest EPE, which reflects that our corrected video has the most promising quality.

Ablation Study

Our network consists of multiple modules. We design several experiments to verify the effectiveness of each key component. The experimental results are shown in Table 2. When the network contains only the flow estimation network, the experimental results are unsatisfactory. A sim-

ple network model does not predict optical flow accurately. When the flow enhancement network is introduced, the network obtains a more accurate optical flow by further perceiving the temporal information and local deformation, thereby improving the performance. Compare the second and third rows, removing the temporal weighting scheme reduces the stability, which explicitly indicates the effectiveness of the scheme. From the second and last two rows, when the temporal deformation aggregator and the spatial deformation perception module are unavailable, the change of the stability is not obvious but the quality performance is greatly degraded. It proves that both temporal information and local deformation benefit optical flow estimation.

Conclusion

In this paper, we design a novel network to correct the fisheye video for the first time. Although the existing methods provide satisfactory results on the image, the problem of jitter will be incurred on the video correction. Thus, we first propose a temporal weighting scheme to alleviate the jitter caused by the new input frames. Then, we introduce a temporal deformation aggregator to establish the temporal correlation of features, thereby improving the global accuracy of the intra-frame optical flow. Besides, our spatial deformation perception module can help the network perceive the deformation of the object and improve the local accuracy of intra-frame optical flow. Compared to the state-of-the-art methods, our proposed method achieves the best performance in terms of correction quality and stability. In the future work, we will improve the method to meet the real-time requirements.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62172032, 62120106009).

References

- Alemánflores, M.; Álvarez, L.; Gómez, L.; and D.SantanaCedrés. 2014. Automatic Lens Distortion Correction Using One-Parameter Division Models. *I POL*, 4: 327–343.
- Barreto, J. 2006. A unifying geometric representation for central projection systems. *Comput. Vis. Image Underst.*, 103: 208–217.
- Basu, A.; and Licardie, S. 1995. Alternative models for fish-eye lenses. *Pattern Recognition Letters*, 16: 433–441.
- Bogdan, O.; Eckstein, V.; Rameau, F.; and Bazin, J. 2018. DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *CVMP*.
- Bukhari, F.; and Dailey, M. N. 2013. Automatic Radial Distortion Estimation from a Single Image. *Journal of Mathematical Imaging & Vision*, 45(1): 31–45.
- Dansereau, D.; Pizarro, O.; and Williams, S. B. 2013. Decoding, Calibration and Rectification for Lenselet-Based Plenoptic Cameras. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1027–1034.
- Girshick, R. B.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.
- Han, H.; Zhou, M.; Shang, X.; Cao, W.; and Abusorrah, A. M. 2021. KISS+ for Rapid and Accurate Pedestrian Re-Identification. *IEEE Transactions on Intelligent Transportation Systems*, 22: 394–403.
- Johnson, J.; Alahi, A.; and Li, F. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*, 694–711.
- Johnson, J.; Hariharan, B.; van der Maaten, L.; Li, F.; Zitnick, C. L.; and Girshick, R. B. 2017. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1988–1997.
- Li, X.; Ma, C.; Wu, B.; He, Z.; and Yang, M. 2019a. Target-Aware Deep Tracking. *CVPR*, 1369–1378.
- Li, X.; Zhang, B.; Sander, P. V.; and Liao, J. 2019b. Blind Geometric Distortion Correction on Images Through Deep Learning. In *CVPR*, 4855–4864.
- Liao, K.; Lin, C.; Zhao, Y.; and Gabbouj, M. 2019. DR-GAN: Automatic Radial Distortion Rectification Using Conditional GAN in Real-Time. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Liao, K.; Lin, C.; Zhao, Y.; and Xu, M. 2020. Model-Free Distortion Rectification Framework Bridged by Distortion Distribution Map. *IEEE Transactions on Image Processing*, 29: 3707–3718.
- Lin, T.-Y.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.
- Lin, T.-Y.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 318–327.
- Liu, K.; Ma, B.; Zhang, W.; and Huang, R. 2015. A Spatio-Temporal Appearance Representation for Video-Based Pedestrian Re-Identification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 3810–3818.
- Liu, S.; Yuan, L.; Tan, P.; and Sun, J. 2013. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32: 1 – 10.
- Luong, Q.-T.; and Faugeras, O. D. 2004. Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. *International Journal of Computer Vision*, 22: 261–289.
- Martin, H.; Hubert, R.; Thomas, U.; Bernhard, N.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NIPS*.
- Mei, C.; and Rives, P. 2007. Single View Point Omnidirectional Camera Calibration from Planar Grids. *IEEE International Conference on Robotics and Automation*, 3945–3950.
- Melo, R.; Antunes, M.; Barreto, J. P.; Fernandes, G. F. P.; and Gonalves, N. 2013. Unsupervised Intrinsic Calibration from a Single Frame Using a “Plumb-Line” Approach. *ICCV*, 537–544.
- Newell, A.; Yang, K.; and Deng, J. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*.
- Pedagadi, S.; Orwell, J.; Velastín, S. A.; and Boghossian, B. A. 2013. Local Fisher Discriminant Analysis for Pedestrian Re-identification. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3318–3325.
- Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Gool, L. V.; Gross, M. H.; and Sorkine-Hornung, A. 2016. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 724–732.
- Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- Reimer, P.; Bard, E.; Bayliss, A.; Beck, J.; Blackwell, P. G.; Ramsey, C.; Buck, C.; Cheng, H.; Edwards, R.; Friedrich, M.; Grootes, P.; Guilderson, T. P.; Haffidason, H.; Hajdas, I.; Hatté, C.; Heaton, T.; Hoffmann, D.; Hogg, A.; Hughen, K.; Kaiser, K.; Kromer, B.; Manning, S.; Niu, M.; Reimer, R.; Richards, D.; Scott, E. M.; Southon, J.; Richard, S.; Turney, C.; and Plicht, J. V. D. 2009. Intcal09 and marine09 radiocarbon age calibration curves, 050,000 years cal bp. *Radiocarbon*, 51: 1111–1150.
- Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39: 1137–1149.

- Rong, J.; Huang, S.; Shang, Z.; and Ying, X. 2016. Radial Lens Distortion Correction Using Convolutional Neural Networks Trained with Synthesized Images. In *ACCV*.
- Sampat, M. P.; Wang, Z.; Gupta, S.; Bovik, A. C.; and Markey, M. K. 2009. Complex Wavelet Structural Similarity: A New Image Similarity Index. *IEEE Transactions on Image Processing*, 18: 2385–2401.
- Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39: 640–651.
- Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, 402–419. Springer.
- Xu, N.; Yang, L.; Fan, Y.; Yang, J.; Yue, D.; Liang, Y.; Price, B.; Cohen, S.; and Huang, T. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 585–601.
- Xue, Z.; N.; Xia, G.; and Shen, W. 2019. Learning to Calibrate Straight Lines for Fisheye Image Rectification. *CVPR*, 1643–1651.
- X. Yin; Wang, X.; Yu, J.; Zhang, M.; Fua, P.; and Tao, D. 2018. FishEyeRecNet: A Multi-context Collaborative Deep Network for Fisheye Image Rectification. In *ECCV*, 475–490.
- Yang, S.; Lin, C.; Liao, K.; Zhang, C.; and Zhao, Y. 2021. Progressively Complementary Network for Fisheye Image Rectification Using Appearance Flow. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6344–6353.
- Zhang, S.; Yao, H.; Sun, X.; and Lu, X. 2013. Sparse coding based visual tracking: Review and experimental comparison. *Pattern Recognition*, 46: 1772–1788.
- Zhang, Z. 1999. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, 1: 666–673 vol.1.
- Zhang, Z. 2000. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22: 1330–1334.
- Zhao, K.; Lin, C.; Liao, K.; Yang, S.; and Zhao, Y. 2021. Revisiting Radial Distortion Rectification in Polar-coordinates: A New and Efficient Learning Perspective. *IEEE Transactions on Circuits and Systems for Video Technology*.