実世界におけるモーション補正魚眼映像強調器

Motion Rectified Fisheye Video Enhancer in Real-world Scenario

張　宇鵬* Yupeng ZHANG　張　恒之* Hengzhi ZHANG　賈　海晶* Haijing JIA　劉　麗艷* Liyan LIU　伊　紅* Hong YI　董　滨* Bin DONG

要　旨

魚眼ビデオの強化は，ローレベルビジョン（low-level vision）タスクではほとんど取り上げられていない．本稿では，1920×1920サイズの魚眼フレーム全体の画像品質を向上させることができる魚眼映像強化器（FVE）を提案する．この目標を達成するために，まず実世界の魚眼・カムコーダーデータセットと，トレーニング用の適切に整列されたパッチペアを生成するためのマッチングアルゴリズムを提案する．取得したデータセット内の大きな動きを処理するために，モーションリファイナー（MR）を提案し，短期・長期融合画像とフローを利用してモーションブラーを修正し，画像のシャープネスを復元する．実験によって，MRモジュールは我々のFVEにとって不可欠な要素であり，CNNまたはトランスフォーマーアーキテクチャに基づく既存の作品よりも優れた品質を達成していることが示された．

ABSTRACT

Fisheye video enhancement is seldom addressed in low-level vision tasks. In this paper, we propose a fisheye video enhancer (FVE) capable of improving overall image quality of 1920×1920-sized fisheye frames. To achieve this goal, we first develop a real-world fisheye-camcorder dataset and a matching algorithm to generate well-aligned patch pairs for training. To tackle large motion in the captured dataset, we also propose a Motion Refiner (MR), which rectifies motion blur and restores image sharpness by using short-long-term fused images and flows. Experiments show that the MR module is an indispensable component in our FVE and yields considerably higher quality images than existing methods based on CNN or Transformer architectures.

* リコーソフトウェア研究センター北京
Ricoh Software Research Center Beijing Co., Ltd.

1. Introduction

Panoramic image and video super-resolution (VSR) have seen great development in recent years[1-3]. Most of the works design panoramic-specific modules in their network architecture and loss, aiming to deal well with severe distortion in the image/frame. These panoramic-specific methods are usually focusing on equirectangular projection (ERP)[1-3], which is transformed from dual fisheye sensor image. There are few works studying fisheye image and video super-resolution according to our survey. Although most consumer fisheye cameras have a built-in fisheye to ERP convertor and directly output ERP image, some applications use customized fisheye lens cameras that do not have such feature. For example, for construction site application, people usually use customized fisheye camera that is suitable for on-site inspection. In this case, a fisheye image/video enhancer that can improve resolution, quality and sharpness of the captured image, as well as remove noise, blur and artifact is highly demanding.

In this paper, we deal with video enhancement for panoramic fisheye videos. Fisheye video is the original sensor signal captured by the fisheye lens camera before projection transformation (e.g., to ERP). Although the perspective view is a user-friendly projection and does not introduce significant distortion, it has limited field of view compared to fisheye projection and ERP. In construction site, it is usually necessary to view the entire wide-open space for monitoring and inspection purposes. In this case, fisheye image or ERP image is better because they have much wider field of view (180° to 360°). More importantly, fisheye image is the native image format captured directly by the fisheye panoramic camera. It possesses consistent pixel-density[3]. Projection transformation such as ERP and perspective break this consistency[3] and introduce processing artifact, which leads to unrealistic image when applying enhancement. Therefore, it is favorable to enhance the original fisheye image than ERP or perspective image. This is the motivation of our work.

One important aspect is the dataset used for training. Existing VSR methods usually train a network using synthetic dataset[4-6], in which bicubic or blur degradation is used to generate low resolution (LR) input. The trained model using synthetic dataset does not generalize well to real-world data because the latter includes much more complicated degradations such as optical aberrations, lens distortion, sensor noise etc. that are difficult to synthesize. In our case, we aim to enhance captured fisheye video frames directly to higher resolution and quality without downsampling. For example, given an input 4K×2K dual fisheye, we enhance it to an output 8K×4K fisheye with higher quality. Therefore, a real-world dataset consisting of original-size fisheye frames as the input and higher resolution and quality counterparts as the ground truth (GT) is required. Since it is non-trivial and expensive to obtain paired input-GT full fisheye videos, we simplify this process by adopting only the central region of the fisheye which is aligned with a higher quality GT counterpart. We show that our model trained on the central region of the fisheye generalizes well to other regions too (Fig. 6), and brings visually appealing full fisheye enhanced frames. To ensure the image quality of the captured GT video, we decide to use a 4K camcorder instead of phone cameras because the former has better optical system and sensor than the latter such that high quality GT videos can be obtained. Based on the above reasons, we propose a real-world fisheye-camcorder dataset and a matching algorithm which can align the fisheye central region with the camcorder's frame, then extract paired input-GT patches from the aligned frames. We then use the extracted patches for training.

Another important aspect is the motion blur in the captured data. We found that it is hard to remove large motion blurs using state-of-the-art (SOTA) VSR or video restoration networks for our dataset. This is probably because the fisheye data is much more complicated than

public datasets used for the SOTAs. The complexity of fisheye data generally lies in three folds: First, fisheye image has much lower quality than ordinary camera's because the former's angular resolution is very low. Second, fisheye motion blur does not simply follow object movement along one dimension but with two dimensions[7]. Third, the captured image itself also embeds with some camera-specific noise which is unique to the fisheye camera. All these add to the difficulty for motion blur removal and image quality improvement for fisheye videos. To recover image sharpness and obtain a perceptually good enhanced fisheye video, we propose the motion rectified fisheye video enhancer, or FVE. The motion rectification is realized by a module inside FVE called Motion Refiner (MR). We will show that the MR is a crucial and indispensable component in our network for blur removal. We further show that an image and flow discriminators following the MR module are also important for perceptual quality improvement.

Our contributions are as follows:

- Propose a fisheye video enhancer (FVE) capable of enhancing full-size fisheye video frame to high resolution and image quality.
- Propose a fisheye(input)-camcorder (GT) real-world panoramic video enhancement dataset (fish-cam dataset in short) and a matching algorithm to align the fisheye central region with the camcorder's frame.
- Propose a motion refiner (MR) to tackle large motions which are difficulty for existing SOTA networks and restore sharpness of the enhanced frames. Our FVE equipped with MR achieves superior performance than existing state-of-the-arts (SOTAs).

2. Related arts

2-1 Panoramic image/video super-resolution

Most of the existing panoramic image or video super-resolution methods focus on module and loss design on ERP based on its pixel density distribution and distortion[1,3,8-10,11]. For example, a latitude adaptive upscaling algorithm for ERP super-resolution was proposed,[8] a distortion aware attention block considering different latitudes on the ERP was also proposed,[3] a dual learning strategy and a weighted loss was adopts to handle non-uniform information distribution on the ERP[1], and a spherical signal super-resolution with proportioned optimization (S3PO) model was designed for ERP VSR.[10] Although fisheye image/frame includes more complicated distortion than ERP[3], we demonstrate a simplified way of training by adopting only the central region of the fisheye which is aligned with a higher quality GT counterpart. Without adaptive design for the fisheye, the trained model is still proved to generalize well to the peripheral regions (Fig. 6).

2-2 Video super-resolution for ordinary camera

A lot of VSR methods focus on the design of temporal alignment and fusion module. EDVR[12] uses pyramid cascading deformable convolution (PCD) for alignment and temporal-spatial attention (TSA) fusion module to emphasize different informativeness of each neighboring frame with respect to the reference frame. BasicVSR[13] uses a bidirectional propagation to strengthen long-term information connection, and emphasizes the importance of feature alignment over image alignment. Its updated version IconVSR[13] adds an information-refill module to mitigate accumulated error in long-term frame alignment. Other works such as deformable convolution[14] and DUF[15] use non-optical-flow-based method for alignment. BasicVSR++[16] adopts a hybrid version using flow-guided

deformable alignment to stabilize training. Recently, a bunch of transformer-based methods[6,17,18,19] have also been proposed. However, none of the above is designed for fisheye video data that are more challenging than the ordinary camera data as we described in section 1.

2-3 Synthetic and real-world dataset

Considering the difficulty of collecting real-world paired data, a lot of VSR works adopt synthetic data in which the degradation is realized by downsampling and/or blurring operation from the GT frames[4,5,13-17,20]. In the panoramic image/video SR field, most of existing methods also use synthetic dataset[1,9,21,22].[1)] proposes a MiG panorama dataset which consists of ERP videos from the Internet. Their training input and GT are downsampled 8× and 2× respectively, directly from the original ERP using bicubic interpolation. Lau-net[8)] generates LR input by bicubic downsampling on the ERP HR image with scaling factor 8× and 16× on the ODI-SR and SUN 360 Panorama[23)] datasets.

Our dataset is different in that the input data are directly extracted from the fisheye image. More importantly, we adopt real-world dataset where the input LR is not downsampled from the GT but captured directly from a dual-fisheye camera. Our GT is captured by a 4K camcorder with much higher quality than our input. Then we directly apply trained model on the full-size fisheye frames of 2K×2K resolution (one fisheye). We are doing this because we aim to enhance captured fisheye video frames directly to higher resolution and quality. Real-world data have the merit of real-world degradation which is much more complicated than synthetic one[24-26], thus a carefully trained model can generalize well to practical applications. Many works capture real-world paired input-GT data by using different cameras or same camera with different focal lengths[26)]. To cope with the different exposure, optical aberrations and color balance in the data, people also use matching algorithm for alignment. Although real-world VSR and dataset have been studied for ordinary cameras e.g., phones' or DSLRs[24)], few has studied panoramic real-world VSR[1)], let alone collecting a corresponding dataset. In this work, we focus on the study of real-world fisheye VSR by presenting a dataset as well as a network that is tailored to tackle the complexity of the dataset.

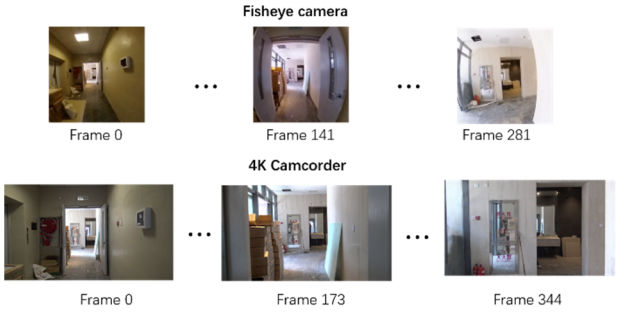


Fig. 1 Top: consecutive frames from one fisheye video clip (only central regions extracted from the full fisheyes are shown); Bottom: consecutive frames from the camcorder at the same scene captured at the same period. Here the camcorder's frame number is 1.223 times that of the fisheye.

3. Methodology

3-1 Real-world fish-cam video enhancement dataset and matching algorithm

In this paper, we improve overall quality of the input fisheye video with a scale factor × 2. More specifically, we super-resolve an input single fisheye frame from resolution 1920×1920 (2K) to 3840×3840 (4K), improve texture details and image clarity, remove blurs, noise and artifact simultaneously. The reason why we only test 2× is that it maintains a good balance between processing time and image quality: lower scale factor (1×) may not produce acceptable perceptual quality, and higher scale factor (e.g., 4× or 8×) is very memory-hungry and takes very long time to process one single fisheye frame because its resolution will become very large: 7680×7680 (in case of SR4×). One challenge is how to align the videos spatially and temporally. Unlike ordinary real- world VSR

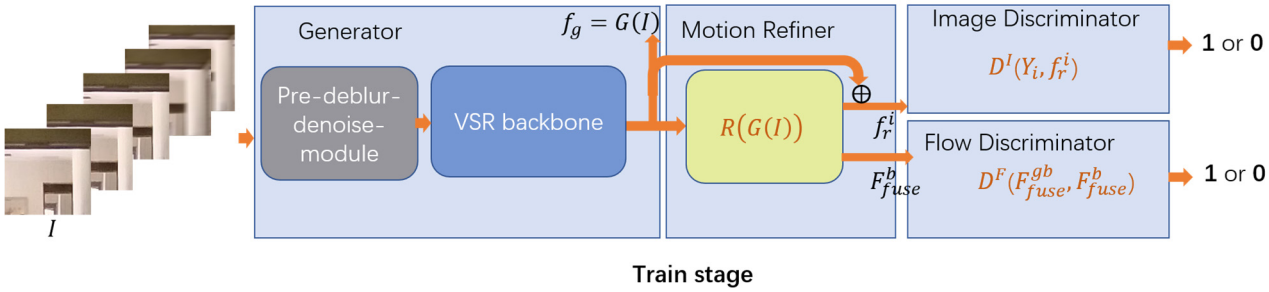


Fig. 2 Our FVE network consists of four main components: generator, motion refiner (MR), image and flow discriminators. For simplicity purpose, only backward flows are shown.

works in which the video pairs of input-GT frames are all captured in the perspective projection[7,27], our fish-cam real-world dataset includes pairs that are in different projections, i.e., fisheye and perspective for input and ground truth, respectively. Unlike the distortion-free ground truth frames, the fisheye frames are severely distorted around image boundaries. A matching algorithm is therefore required to precisely align the video pairs (spatially alignment). Since the field of view (FOV) of our camcorder frames are mostly overlapped with the central region of our fisheye frames, our algorithm can be simplified by matching only the overlapped region where the distortion is not so severe. Additionally, the fisheye camera and the camcorder have different frame rate, resulting in different number of frames even if captured in the same periods of time. Therefore, a temporal alignment method should also be considered. We summarize our data collection and matching algorithm as follows:

- Capture low-quality videos using the fisheye camera and the corresponding ground truth videos using the 4K camcorder.
- Select a set of continuous frames from each type of videos. Figure 1 shows two sets of frames recorded at the same scene.
- Adjust the frame number of the two sets for temporal alignment: to ensure that the camcorder's frame number is N times that of the fisheye (in our case the former's frame rate is a little bit higher), where N is not necessarily integer. Then we use the int(i×N)th frame of the camcorder to spatially align with the i th frame of the fisheye, where int indicates integer operation. The spatial alignment further consists of three steps: a) coarse matching, b) fine matching and c) template matching.

One issue caused by the real-world dataset captured by two different cameras is the misalignment of LR-GT frame pixels and color/luminance difference[24]. To mitigate the influence of these factors on model performance, we can either combine synthetic datasets (e.g., REDS[28]) with the fish-cam dataset or initialize a training using a pre-trained model of the combined dataset. As we mentioned in section 2.3, the LR frames of the synthetic dataset is generated by downsampling the GT. There is no pixel misalignment and color difference issues which is common in the real-world dataset. The inclusion of synthetic dataset helps improving model performance and reducing color/luminance difference between the input and output frames.

3-2 Network architecture

Our network has four main components: a generator, a motion refiner, an image discriminator and a flow discriminator. The generator further includes two modules: a pre-deblur-denoise module to remove noise and blur before feeding frames to the next module, a VSR backbone module for frame propagation, alignment and aggregation. The proposed motion refiner (MR) aims to rectify large motions that are unable to be fully resolved

by the generator. The image discriminator aims to generate photorealistic output frame from the MR module. The flow discriminator aims to generate fused short-long-term flows as close to the GT flows as possible. The full architecture is shown in Fig. 3.

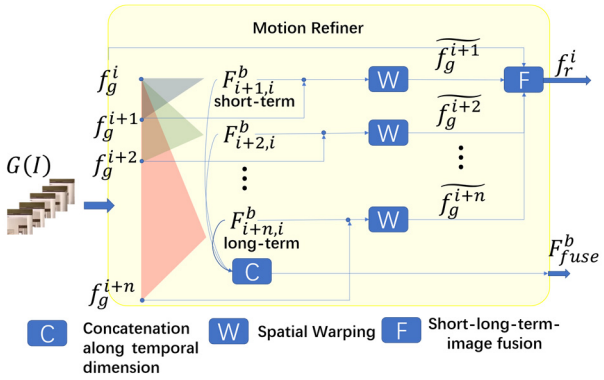


Fig. 3 Motion Refiner. Only backward flows are shown for simplicity purpose.

As mentioned in section 1, our dataset is complicated in terms of image quality, noise and motion blur. An effective pre-deblur-denoise module can alleviate the burden of the subsequent VSR backbone network. We adopt the multi-frequency RRDB structure introduced in[29] to initially remove noise and blur from the input frames. Then, the VSR backbone performs the basic tasks of frame propagation, alignment and aggregation, which are widely used in most VSR networks. The backbone can be any SOTA network which is insertable between the pre-deblur-denoise module and the MR. Specifically, an input sequence denoted as I with shape (b, t, c, h, w) is fed to the generator G and output an enhanced sequence $f_g = G(I)$ with shape $(b, t, c, 2h, 2w)$. Here b, t, c, h and w represent batch size, number of neighboring frames, number of input and output channels, and height and width of one frame, respectively. The generator output $G(I)$ serves as the input of the subsequent MR.

In the following paragraphs, we mainly discuss the proposed motion refiner and the subsequent two discriminators.

The MR module aims to rectify motion blurs which are hard to be removed by the generator alone. A rational way to achieve this is to examine both adjacent and distant optical flows of a blurry frame from $G(I)$ and use precisely estimated flows to warp towards this frame. The reason for incorporating both adjacent and distant flows is that the adjacent frames of the blurry one are usually blurry too, in which case the accurate flow cannot be estimated due to intensity change and mis-matching[30]. The distant frames, however, are highly possible to include sharp ones, which can be used to compute accurate flows. In the MR module, we concatenate distant flows with adjacent ones along the temporal dimension and feed them to the subsequent flow discriminator. Each adjacent or distant flow is also used to warp towards the blurry frame, and the original blurry frame and warped ones are fused together, then fed to the subsequent image discriminator. These short-long-term flow and image fusions help refining motion blurs that cannot be fully resolved by the generator.

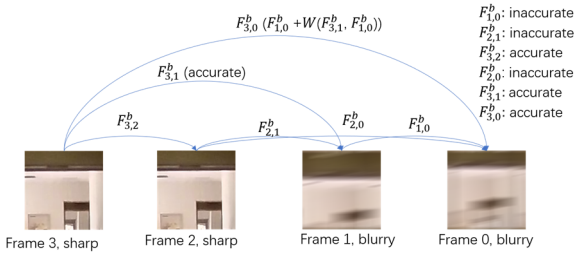


Fig. 4 Residual short-long-term flow computation. The long-term flows are useful to estimate accurate optical flows when they are computed from sharp frames.

Specifically, as shown in Fig. 3, let f_g^i be an arbitrary blurry frame from $G(I)$ where $i \in (1,2,\dots,t)$. Its neighboring frames are denoted as $f_g^{i+1}, f_g^{i+2}, \dots, f_g^{i+n}$. For simplicity purpose, we only consider backward flows in the following definitions. Then, we can compute adjacent and distant backward flows as $F^b_{i+1,i}, F^b_{i+2,i}, \dots, F^b_{i+n,i}$. Here we adopt residual flow defined as follows for flow computation from frame $i+n$ to i

$$F^b_{i+n,i} = F^b_{i+1,i} + W\ (F^b_{i+n,i+1},\ F^b_{i+1,i}) \quad (1)$$

where the flow from frame $i+n$ to $i+1$, i.e., $F^b_{i+n,i+1}$ are warped towards flow $F^b_{i+1,i}$, i.e., flow from frame $i+1$ to i, then added with $F^b_{i+1,i}$. The distant flow $F^b_{i+n,i+1}$ can be computed using the same definition in (1). An illustration of residual flow between four frames is shown in Fig. 4. As to adjacent flows in Fig. 4, only $F^b_{3,2}$ is accurate because frame 2 and 3 are sharp. This flow map can be used to compute distant flows $F^b_{3,1}$ and then $F^b_{3,0}$ using (1), which are also accurate. As a result, the distant flow $F^b_{3,0}$ is used in the warping module to warp frame 3 towards 0, obtaining a sharp frame.

The adjacent and distant flows are then passed to the warping module of MR, which warp the corresponding frames toward f^i_g. This can be expressed as follows:

$$\widetilde{f^{i+n}_g} = W(f^{i+n}_g, F^b_{i+n,i}) \quad (2)$$

where W indicates spatial warping operation that warps frame $i+n$ towards i and $\widetilde{f^{i+n}_g}$ is the warped frame. In this way, we can obtain warped frames $\widetilde{f^{i+1}_g}$, $\widetilde{f^{i+2}_g}, \dots, \widetilde{f^{i+n}_g}$ that may include sharp ones, as shown in Fig. 3. We then fuse the warped frames with f^i_g to obtain f^i_r, which is one output of the MR module. This can be defined as

$$f^i_r = F(f^i_g, \widetilde{f^{i+1}_g}, \widetilde{f^{i+2}_g}, \dots, \widetilde{f^{i+n}_g}) + f^i_g \quad (3)$$

where F indicates short-long-term image fusion along the channel dimension followed by some convolutional layers. Note that f^i_r here only includes fused frames from the backward direction. The full f^i_r should also include frames from the forward direction, which is straightforward to add. f^i_r serves as the input of the subsequent image discriminator.

On the other hand, the obtained short-long-term flows $F^b_{i+1,i}, F^b_{i+2,i}, \dots, F^b_{i+n,i}$ are concatenated along the temporal dimension to obtain a fused backward flow map F^b_{fuse}.

$$F^b_{fuse} = C(F^b_{i+1,i}, F^b_{i+2,i}, \dots, F^b_{i+n,i}) \quad (4)$$

where C means concatenation along the temporal dimension. Similarly, we can obtain the fused forward flows as F^f_{fuse}. The fused flows F^b_{fuse} and F^f_{fuse} are the other two outputs of the MR and serve as the inputs of the subsequent flow discriminator.

Overall, our MR module denoted as R can be defined as follows:

$$f^i_r, F^b_{fuse}, F^f_{fuse} = R(G(I)) \quad (5)$$

The initial backward and forward flows between adjacent frames are computed using pretrained Spynet[31] model, after which all parameters inside R are updated for each iteration. The shape of f^i_r is $(b,3,2h,2w)$ and the shapes of F^b_{fuse} and F^f_{fuse} are $(b,f_{sl},2,2h,2w)$, where f_{sl} indicates the number of concatenated short-long-term flows for each of the backward and forward flows. Here the channel number equals 2 indicating the vertical and horizontal components of each flow.

The image and flow discriminators in our network aim to generate indistinguishable fused short-long-term images f^i_r and flows F^b_{fuse}, F^f_{fuse} from their corresponding ground truth versions. We use relativistic discriminator[32] to achieve the target. Specifically as shown in Fig. 2, we denote $D^I(Y_i, f^i_r)$ as the image discriminator, where $Y_i (i \in 1,2,\dots,t)$ is the ith GT frame corresponding to f^i_r. Similarly, the flow discriminator is denoted as $D^F(F^{gb}_{fuse}, F^b_{fuse})$ and $D^F(F^{gf}_{fuse}, F^f_{fuse})$, where F^{gb}_{fuse} and F^{gf}_{fuse} are the corresponding fused GT flows for backward and forward directions, respectively. Note that F^{gb}_{fuse} and F^{gf}_{fuse} follow the same equations (1) and (4) to compute adjacent and distant flows as well as concatenation. The output of D^I and D^F are either 1 or 0, indicating more realistic than a fake data or less realistic than a real data, respectively. For instance, $D^I \to 1$ means Y_i is more realistic than f^i_r.

Our objective function for generator and motion refiner can be expressed as the following general form:

$$L_{G,R} = \sum_{i=1}^{K} \alpha_i L^{I,F}_{G,R} \quad (6)$$

Note that $L_{G,R}$ is defined across both G and R nets. The subscript in (6) indicates the loss is computed from output of either G or R. The superscript I or F means loss computed for image or flow, respectively. α_i is the

(a) Input frame (b) Input close-up (c) BasicVSR[13] (d) BasicVSR++[16] (e) VRT[17] (f) FVE (Ours)

Fig. 5 Central region enhancement. All models are trained using fish-cam dataset and GAN.

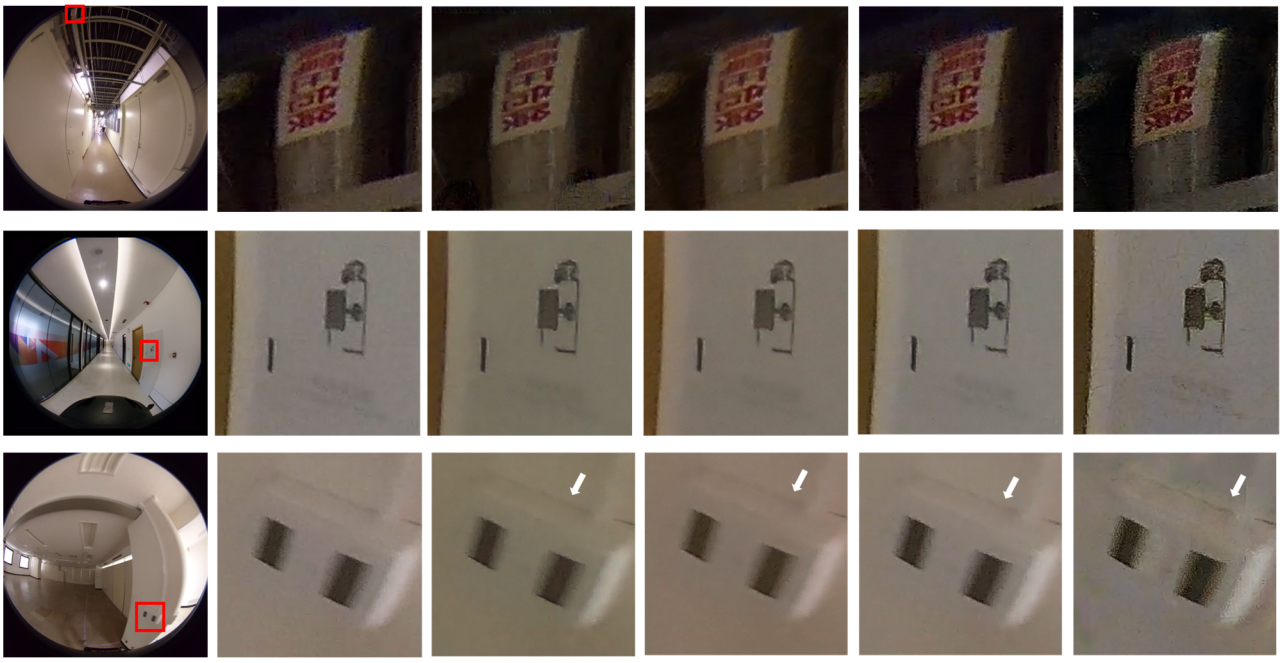
weight for balancing each loss term. In order to obtain perceptually good output frame with rectified motion blur, we let $L_{G,R}^{I,F}$ include pixel-based content loss, perceptual loss in VGG space, relativistic discriminator loss (for both fused image and flows) and end-point-error loss[30].

One thing needs to be mentioned is that we adopt GAN[33] and adversarial losses in our network design. We found that a discriminator as well as GAN losses are necessary because it generates perceptually better results with finer details and sharper edges than non-GAN methods trained with only pixel level loss, e.g., Charbonnier loss[12,13,16,17] or L1 loss[4]. Existing methods seldom use GAN for video enhancement because GAN is difficult to train and not stable. It is good enough to train a non-GAN model for most of the public VSR datasets e.g., REDS, Vimeo[34]. However, in our case, non-GAN model does not perform good because our dataset is much more complicated as described in section 1.

4. Experiments

In this section, we compare our FVE with some VSR methods that are widely adopted in the literature. The methods include BasicVSR[13], BasicVSR++[16] and VRT[17]. Compared to these methods, FVE considers the accuracy of the optical flow estimation by residual short-long-term flow computation introduced in Fig. 4. Besides, FVE is based on pure convolutional network, so it is computationally efficient than an ordinary transformer architecture in vision tasks such as image/video SR. In addition, the proposed FVE adopts a flow discriminator which will generate indistinguishable flows from their corresponding ground truth. None of the aforementioned methods adopt the flow discriminator.

Since we have the aligned camcorder frames by using matching algorithm introduced in section 3.1 (but corresponding to only a small central region in the fisheye frame because the camcorder has limited FOV), we regard them as the GT and evaluate different methods using full-



(a) Input frame (b) Input close-up (c) BasicVSR[13] (d) BasicVSR++[16] (e) VRT[17] (f) FVE (Ours)

Fig. 6 Full-size single fisheye enhancement. All models are trained using fish-cam dataset and GAN.

reference metrics: PSNR, SSIM and LPIPS[35]. Since different metric has different definition and focus for evaluating image quality, we believe that these full-reference metrics can reflect the overall performance of the model. For full-fisheye images, we cannot use full-reference metrics because we have no corresponding full-fisheye GT. In this case, we evaluate using no-reference metrics: NIQE[36], and BRISQUE[37].

On the other hand, the evaluation of our method cannot apply WS-PSNR/WS-SSIM that are designed for EPR image/frames[38,39] because fisheye possesses different pixel density distribution with ERP. To compare fairly with the existing methods, we adopt non-panoramic methods such as EDVR, VRT etc., and train with GAN (only image discriminator) using the same losses as ours. Some methods output multiple frames instead of single frame. In that case, we feed the discriminator with the central output frame only.

Table 1 Full-reference evaluation on fisheye central region. All scores are averaged over 300 enhanced images of size 1280×640 by referring to the corresponding GT of the same size. PSNR and SSIM are computed for RGB channels. Best marked in red.

	PSNR (↑)	SSIM (↑)	LPIPS (↓)
BasicVSR[13]	15.02	0.6625	0.5491
BasicVSR+[16]	14.92	0.6631	0.5352
VRT[17]	14.52	0.6570	0.5389
Ours (FVE)	15.46	0.6461	0.5188

As mentioned in 3.1, we combine synthetic dataset with our fish-cam dataset to mitigate the influence to the model performance caused by the misalignment of input-GT frames. Specifically, we add REDS to our fish-cam dataset with a ratio of training pairs 1:1. Note that all SOTAs and our model are trained using the same fish-cam dataset combined with REDS. The full-reference metrics are computed between enhanced frames by different methods

and the GT. All enhanced frames are super-resolved with scaling factor 2×.

Table 1 and Fig. 5 show full-reference results and the corresponding visual quality of all compared methods for central region of the fisheye frame. We can see that our FVE are very effective for blur removal in all three sample images, which all possess motion blurs of some extent. Especially, the third row shows large motion blur around the left and right parts of the image. Thanks to the proposed MR module, our method can recover sharp edges and clear textures. Other methods also improve the sharpness and clarity over the input image, but they are not comparable to our results. Our FVE also outperforms other compared methods for PSNR and LPIPS scores with large margin.

Table 2 and Fig. 6 show no-reference results and the corresponding full-fisheye visual results. In Fig. 6, we show enhanced regions in the peripheral areas to validate that our model trained on the central region also performs well on other areas. Our visual results outperform other compared methods in terms of image sharpness and texture improvement (see the object edge marked with arrows on the cropped area in the third row). The quantitative results in Table 2 also show that our FVE is superior to other methods in terms of image quality evaluated by no-reference metrics.

Table 2 No-reference metrics: BRISQUE and NIQE are computed by averaging 180 full single fisheye enhanced images selected from six video clips. Best marked in red.

	BRISQUE(↓)	NIQE(↓)
BasicVSR[13)]	27.55	4.6229
BasicVSR++[16)]	24.24	4.9305
VRT[17)]	32.76	5.7020
Ours (FVE)	23.4	3.2981

In addition, we also show the processing time of compared methods in Table 3. We tested three methods using 2K (1920×1920) fisheye frames and generate 4K (3840×3840) super-resolved frames. The input frame number is set to 9, which means all nine neighboring frames are used to compute optical flow and fed to the motion refiner. We run all methods on a single RTX 4070Ti Super GPU to make sure fairness comparison. The processing time includes only the inference time from loading the model to output the result. For FVE, we adopt BasicVSR as the VSR backbone (please refer to Fig. 2 for more details). We compute average processing time over 30 consecutive frames.

Table 3 Processing time comparison.

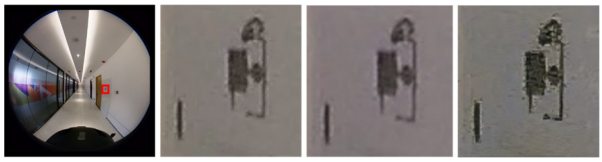
Methods	Average processing time(s)
BasicVSR[13)]	24.31
VRT[17)]	3583.55
Ours (FVE)	38.19

We can see that the processing time for FVE is approximately 1.57 times that of BasicVSR, and much faster than VRT, which runs about 1 hour. The full and no reference scores in Tab.1 and 2 show that FVE surpasses BasicVSR and VRT with a large margin. This indicates that enhancement by scale factor 2× maintains a good balance between processing speed and image quality.

Finally, to investigate the effectiveness of the proposed real-world dataset, we compare model performance on a synthetic dataset and our fish-cam dataset. We used ODV360 dataset[40)] as the synthetic data for training. Since ODV360's original HR frames format is EPR with resolution 2160×1080, we convert them to fisheye frames first. Each ERP frame is converted to left and right fisheyes with resolution 1080×1080. Then we downsample the left and right fisheye by scale factor 2× to create the LR frames. EDVR was adopted as the VSR backbone. To compare with model trained with our fish-cam real-world dataset, we tested both models on real-world data. Quantitative and qualitative results are shown in the Table 4 and Fig. 7. For quantitative evaluation, we use no-reference metrics BRISQUE and NIQE for 30 consecutive

frames and compute the average scores as the final scores. For qualitative evaluation, we denote model trained by ODV360 dataset as Model ODV360 and those by real-world dataset fish-cam as Model Fish-Cam.

Note that model trained using synthetic dataset generates enhanced image with poor perceptual quality. This is because the model is trained using pure synthetic data which cannot generalize well to the real-world data.



(a) (b) (c) (d)

Fig. 7 Qualitative comparison between model trained by ODV360 (synthetic) and fish-cam (real-world) dataset. (a) Input frame. (b) Input close-up. (c) Model ODV360. (d) Model Fish-Cam.

Table 4 Quantitative comparison between model trained by ODV360 and Fish-Cam dataset. Scores are averaged over 30 consecutive fisheye frames.

	ODV360	Fish-Cam
BRISQUE(↓)	43.44	23.4
NIQE(↓)	4.2875	3.2981

5. Ablation study

To further verify the effectiveness of the proposed FVE, and the necessity of the MR module and subsequent image and flow discriminators, we carried out ablation study by adding or removing important modules. We compare our full model using all four components: generator $\boldsymbol{G}$, motion refiner $\boldsymbol{R}$, image discriminator $\boldsymbol{D^I}$ and flow discriminator $\boldsymbol{D^F}$ with model using only $\boldsymbol{G}$ and $\boldsymbol{D^I}$. Table 5 shows the full and no reference scores and Fig. 8 shows the visual results. Note that the first column of Fig. 8 shows the close-up of the central region shown in Fig. 5.

We can see that the model using only $\boldsymbol{G}$ and $\boldsymbol{D^I}$ cannot effectively remove large motion blur shown in the third row of Fig. 8 (also refer to Fig. 5 third row for large motion of this area) because it lacks the MR module which rectify large motions that $\boldsymbol{G}$ alone cannot resolve. For texture improvement shown in the first and second rows, $\boldsymbol{GD^I}$ also shows unsatisfied results. Our full model $\boldsymbol{GRD^ID^F}$(FVE) significantly improves the image quality over $\boldsymbol{GD^I}$, removes large motion blur effectively and recover fine textures as shown in Fig. 8. The quantitative results in Table 5 also show the superiority of the full model over $\boldsymbol{GD^I}$. This suggested that the proposed MR module and the subsequent image and flow discriminators are crucial and indispensable designs in our network.

(a) Input close-up (b) GD^I (c) GRD^ID^F(FVE)

Fig. 8 Ablation study. Visual comparison between our full model GRD^ID^F with the model using only G and D^I.

Table 5 Ablation study. All scores are computed using the same rules as Table 1 and 2. Best marked in red.

	$\boldsymbol{GD^I}$	$\boldsymbol{GRD^ID^F}$**(FVE)**
PSNR(↑)	14.62	15.46
SSIM(↑)	0.6401	0.6461
LPIPS(↓)	0.5242	0.5188
BRISQUE(↓)	27.01	23.40
NIQE(↓)	4.8595	3.2981

6. Conclusion

This paper proposed a fisheye video enhancer (FVE) that enhances full-size fisheye frame to high resolution and image quality. We also proposed a real-world fisheye-camcorder dataset and a matching algorithm to train a model adaptable to real-world degradation. To deal with the complexity of the dataset and large motions that cannot be resolved by the generator alone, we further propose a motion refiner followed by one image and one flow discriminator. Experiment shows the proposed FVE equipped with MR achieves superior quality over compared works trained using GAN and same losses.

References

1) H. Liu et al.: A single frame and multi-frame joint network for 360-degree panorama video super-resolution, arXiv e-print arXiv:2008.10320 (2020).
2) C. Ozcinar, A. Rana, A. Smolic: Super-resolution of omnidirectional images using adversarial learning, In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1-6 (2019).
3) F. Yu et al.: OSRT: Omnidirectional image super-resolution with distortion-aware transformer, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13283-13292 (2019).
4) M. Haris, G. Shakhnarovich, N. Ukita: Recurrent back-projection network for video super-resolution, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3897-3906 (2019).
5) J. Cao et al.: Video Super-Resolution Transformer, *Computer Research Repository (CoRR)*, Vol. abs/2106.06847 (2021).
6) H. Chen et al.: Pre-trained image processing transformer, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299-12310 (2021).
7) A. Eichenseer, M. Bätz, A. Kaup: Motion estimation for fisheye video with an application to temporal resolution enhancement, IEEE Transactions on Circuits and Systems for Video Technology, 29(8), pp. 2376-2390 (2018).
8) X. Deng et al.: Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9189-9198 (2021).
9) Y. Yoon et al.: SphereSR: 360° Image Super-Resolution with Arbitrary Projection via Continuous Spherical Image Representation, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5677-5686 (2022).
10) A. A. Baniya et al.: Omnidirectional Video Super-Resolution using Deep Learning, *IEEE Transactions on Multimedia* (2023).
11) S. Li et al.: Panoramic image quality-enhancement by fusing neural textures of the adaptive initial viewport, In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 816-817 (2020).
12) X. Wang et al.: Edvr: Video restoration with enhanced deformable convolutional networks, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1954–1963 (2019).
13) K. C. Chan et al.: Basicvsr: The search for essential components in video super-resolution and beyond, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4947-4956 (2021).

14) Y. Tian et al.: TDAN: Temporally deformable alignment network for video super-resolution, arXiv preprint arXiv:1812.02898 (2018).
15) Y. Jo et al.: Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3224-3232 (2018).
16) K. C. Chan et al.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5972-5981 (2022).
17) J. Liang et al.: Vrt: A video restoration transformer, arXiv preprint arXiv:2201.12288 (2022).
18) J. Liang et al.: Swinir: Image restoration using swin transformer, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1833-1844 (2021).
19) Z. Wang et al.: Uformer: A general u-shaped transformer for image restoration, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17683-17693 (2022).
20) T. Xue et al.: Video enhancement with task-oriented flow, *International Journal of Computer Vision*, Vol. 127, pp. 1106-1125 (2019).
21) V. Fakour-Sevom, E. Guldogan, J. K. Kämäräinen: 360 panorama super-resolution using deep convolutional networks, In *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, Vol. 1, p. 1 (2018).
22) A. Nishiyama, S. Ikehata, K. Aizawa: 360 single image super resolution via distortion-aware network and distorted perspective images, In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1829-1833 (2021).
23) J. Xiao et al.: Recognizing scene viewpoint using panoramic place representation, In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2695-2702 (2012).
24) X. Yang et al.: Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4781-4790 (2021).
25) J. Cai et al.: Ntire 2019 challenge on real image super-resolution: Methods and results, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2211-2223 (2019).
26) J. Cai et al.: Toward real-world single image super-resolution: A new benchmark and a new model, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3086-3095 (2019).
27) S. K. Nayar: Catadioptric omnidirectional camera, In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 482-488 (1997).
28) S. Nah et al.: NTIRE 2019 challenge on video deblurring and superresolution: Dataset and study, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshop* (2019).
29) Y. Zhang et al.: Toward real-world panoramic image enhancement, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 628-629 (2020).
30) W. Li et al.: Blur robust optical flow using motion channel, *Neurocomputing*, 220, pp. 170-180 (2017).
31) A. Ranjan, M. J. Black: Optical flow estimation using a spatial pyramid network, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4161-4170 (2017).

32) A. Jolicoeur-Martineau: The relativistic discriminator: a key element missing from standard GAN, arXiv preprint arXiv:1807.00734 (2018).
33) I. Goodfellow et al.: Generative adversarial nets, *Advances In Neural Information Processing Systems*, Vol. 27 (2014).
34) T. Xue et al.: Video enhancement with task-oriented flow, *IJCV* (2019).
35) R. Zhang et al.: The unreasonable effectiveness of deep features as a perceptual metric, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586-595 (2018).
36) A. Mittal, R. Soundararajan, A. C. Bovik: Making a "completely blind" image quality analyzer, *IEEE Signal Processing Letters*, Vol. 20, No. 3, pp. 209-212 (2012).
37) A. Mittal, A. K. Moorthy, A. C. Bovik: No-reference image quality assessment in the spatial domain, *IEEE Transactions on Image Processing*, Vol. 21, No. 12, pp. 4695-4708 (2012).
38) Y. Zhou et al.: Weighted-to-spherically-uniform SSIM objective quality evaluation for panoramic video, In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pp. 54-57 (2018).
39) Y. Sun, A. Lu, L. Yu: Weighted-to-spherically-uniform quality evaluation for omnidirectional video, *IEEE Signal Processing Letters*, Vol. 24, No. 9, pp. 1408-1412 (2017).
40) M. Cao et al.: Ntire 2023 challenge on 360deg omnidirectional image and video super-resolution: Datasets, methods and results, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1731-1745 (2023).