Real-RawVSR: Real-World Raw Video Super-Resolution with a Benchmark Dataset

Huanjing Yue, Zhiming Zhang, and Jingyu Yang*

School of Electrical and Information Engineering, Tianjin University, Tianjin, China
{huanjing.yue, zmzhang, yjy}@tju.edu.cn

Abstract. In recent years, real image super-resolution (SR) has achieved promising results due to the development of SR datasets and corresponding real SR methods. In contrast, the field of real video SR is lagging behind, especially for real raw videos. Considering the superiority of raw image SR over sRGB image SR, we construct a real-world raw video SR (Real-RawVSR) dataset and propose a corresponding SR method. We utilize two DSLR cameras and a beam-splitter to simultaneously capture low-resolution (LR) and high-resolution (HR) raw videos with 2×, 3×, and 4× magnifications. There are 450 video pairs in our dataset, with scenes varying from indoor to outdoor, and motions including camera and object movements. To our knowledge, this is the first real-world raw VSR dataset. Since the raw video is characterized by the Bayer pattern, we propose a two-branch network, which deals with both the packed RGGB sequence and the original Bayer pattern sequence, and the two branches are complementary to each other. After going through the proposed co-alignment, interaction, fusion, and reconstruction modules, we generate the corresponding HR sRGB sequence. Experimental results demonstrate that the proposed method outperforms benchmark real and synthetic video SR methods with either raw or sRGB inputs. *Our code and dataset are available at https://github.com/zmzhang1998/Real-RawVSR.*

Keywords: Real-RawVSR, Raw video, Co-Alignment, Bayer pattern

1 Introduction

Capturing images (videos) with a short-focus lens can enlarge the view angles by sacrificing the resolutions while capturing with a long-focus lens can increase the resolutions by sacrificing the view angles. Image (video) super-resolution (SR) is an effective way to get both wide angle and high-resolution (HR) images (videos). Video SR reconstructs an HR video from a low-resolution (LR) input by exploring the spatial and temporal correlations of the input sequence. In recent years, the development of video SR has shifted from traditional model-driven to deep learning based methods [9,31,32,36].

* This work was supported in part by the National Natural Science Foundation of China under Grant 62072331. *Corresponding author: Jingyu Yang.*

The performance of these deep learning based SR methods heavily depends on the training datasets. Considering that the synthetic LR-HR datasets, such as DIV2K [3] and REDS [28], cannot represent the degradation models between real captured LR images and HR images, many real SR datasets are constructed to boost the real-world SR performance. However, most of these datasets are for static LR-HR images, such as RealSR [8] and ImagePairs [18]. Recently, Yang *et al.* [37] proposed the first real-world video SR dataset via capturing with a multi-camera system of iPhone 11 Pro Max. However, the parallax between the LR and HR cameras increased the difficulty for alignment and there are only 2× LR-HR sequence pairs in this dataset due to the limited focal lengths of phone cameras.

On the other hand, there is a trend to utilize raw images for real-scene image (video) restoration, such as low light enhancement [13,14], denoising [1,2,5,23,33,38], deblurring [22], and super-resolution [35,39]. The main reason is that raw images have wide bit depths (12 or 14 bits), *i.e.,* containing the most original information, and its intensity is linear to the illumination. However, there is still little work exploring raw video SR. Liu *et al.* [24] proposed a raw video SR dataset by synthesizing LR raw frames by downsampling from the captured HR raw frames. Even though, there is still a gap between the synthesized LR raw frames and real captured ones, which makes the SR models trained on synthesized data cannot generalize well to real scenes.

Based on the above observations, we propose to construct a real-world raw video SR dataset to facilitate the raw VSR research. Specifically, we build a two-camera system with a beam-splitter to make sure that there is no parallax between the two cameras. In addition, we perform alignment on the captured LR-HR pairs to make them aligned. On the other hand, the current VSR methods [9,32] are mostly based on sRGB frame inputs and the network design for raw sequence inputs has not been well explored. Therefore, we propose a raw VSR network tailored for raw inputs. Specifically, the raw frames are fed into the network in two forms. One is in its original Bayer pattern and the other is in the packed sub-frame version, namely that RGGB pixels are packed into four channels. The features from the two branches are co-aligned, interacted, and fused together to reconstruct the HR sRGB frame. In brief, our contributions can be summarized as follows.

- We construct the first aligned raw VSR dataset for real scenes, which contains LR-HR pairs for 2×, 3×, and 4× magnification in both raw and sRGB domains. By utilizing a beam splitter in our capturing system, we obtain LR-HR pairs without parallax. There are totally 450 video pairs and each video contains about 150 frames.
- We propose a novel raw VSR network by utilizing the raw frames in terms of the original Bayer pattern and its corresponding packed sub-frame pattern. Specifically, we propose co-alignment, interaction, and fusion modules to take advantage of the complementary information from the two branches.
- We introduce a simple but effective color correction method (*i.e.,* channel-based correction), which is beneficial for training with image pairs having

color differences. Experimental results demonstrate that our method outperforms benchmark VSR methods in both sRGB and raw domains.

2 Related Work

2.1 Image and Video SR Datasets

Image SR datasets. The early image SR datasets usually synthesize LR images from the captured HR ones via bicubic downsampling, such as DIV2K dataset [3]. Considering the domain gap between synthesized and real captured LR images, many real-world SR datasets are constructed. For example, the City100 [12] and RealSR [8] datasets, which are captured with different focal length cameras, contain LR-HR pairs in sRGB domain. Zhang *et al.* claimed that using sRGB images to train the SR model is inferior to that trained by raw data [39]. Therefore, they constructed the first SR-Raw dataset for real-world computational zoom. Meanwhile, Xu *et al.* constructed a synthesized raw image dataset for raw image SR [35]. Hereafter, the ImagePairs dataset [18] is constructed by introducing a beam splitter into the capturing system, which enables them to capture a much larger dataset with LR-HR pairs in both raw and sRGB domains. These datasets have greatly promoted the performance of real image SR and laid the foundation for the construction of VSR datasets for real scenes.

Video SR Datasets. Similar to the development of image SR dataset, the video dataset is also shifted from the synthesized ones (such as REDS [28] and Vimeo-90k [36]) to real captured ones (such as RealVSR [37] and BurstSR dataset [1] [4]), from sRGB domain [36,37] to raw domain [4,24]. The RealVSR [37] dataset is constructed by capturing with two different focal length cameras in iPhone 11 Pro Max and the DoubleTake App. Since the focal lengths are limited for phone cameras, there are only 2× LR-HR sequence pairs in this dataset. Recently, RealBasicVSR [11] built a VideoLQ dataset to assess the generalize ability of real-world VSR methods. Since there are no ground truths for these videos, this dataset cannot be used for supervised training.

Inspired by the success of raw image SR, Bhat *et al.* constructed a BurstSR dataset [4] in the raw domain by capturing the burst LR raw images with a phone camera and the HR sRGB images with a DSLR camera. Liu *et al.* constructed the RawVD dataset [24] for videos, which synthesized the LR raw sequences from the captured HR raw sequences via a degradation model. However, as demonstrated in [4], a network trained with synthetic data is expected to have suboptimal performance when applied to real images. Therefore, we propose to construct a Real-RawVSR dataset by capturing real raw sequences with both short and long focal length cameras for different scaling factors, thus providing a real benchmark for raw VSR model training and evaluation.

[1] Since burst image SR is similar to video SR, we present them here other than in the image SR.

2.2 Image and Video SR Methods

SR Methods for Synthesized Data. In the literature, most SR methods are designed based on the synthesized LR-HR pairs. For image SR, most works explore efficient modules to explore spatial correlations, such as the residual channel attention block in RCAN [40], the holistic attention block in HAN [29]. For video SR, both spatial and temporal correlations are essential for SR performance. Therefore, many methods focus on the alignment strategy, such as the optical flow based [6,19,36] and the deformable convolution [15] based, *e.g.* TDAN [31], EDVR [32]. Recently, BasicVSR [9] and its enhanced versions , *i.e.,* IconVSR [9] and BasicVSR++ [10] have achieved superior SR performance by combining forward and backward bidirectional propagation information and optical flow based feature alignment. Hereafter, Zhou *et al.* proposed an effective iterative alignment algorithm and an efficient adaptive reweighting strategy to better utilize the temporal correlations [41].

SR Methods for Real Captured Data. Different from synthesized LR-HR pairs, there are usually spatial misalignment, color mismatching, and intensity variance in the real captured LR-HR pairs. Therefore, the SR methods for real data focus on dealing with these misalignments. Zhang *et al.* introduced the contextual bilateral loss to deal with the spatial misalignment [39], and Cai *et al.* proposed a Laplacian pyramid based kernel prediction network since the real degradation kernels are naturally non-uniform [8]. Besides, the NTIRE challenge on real-world image SR further boosts the SR performance [7,26]. Compared with real image SR, there is a few research on real VSR. RealVSR [37] proposed a Laplacian pyramid based loss to deal with the misalignment and color differences between the LR-HR frames. Considering that in-the-wild degradations could be exaggerated during temporal propagation, RealBasicVSR [11] proposed a pre-cleaning module to reduce noise and artifacts prior to temporal propagation.

SR Methods for Raw Images and Videos. The above methods are generally designed for sRGB images. For raw input SR, the network needs to simultaneously deal with both ISP and SR tasks. The work in [39] directly maps the raw input to an sRGB output via a ResNet. Different from it, Xu *et al.* proposed a dual CNN, where one branch is used for structure reconstruction and the other branch is for color restoration with the LR sRGB image as guidance [35]. Following it, the RawVSR method [24] also utilizes two branches for both detail and color reconstruction. However, the raw LR frames are synthesized.

To our knowledge, there is still no work exploring real-world raw VSR methods and the network design for raw sequence input has not been well explored. In this work, we propose a two-branch interaction network tailored for raw sequence inputs and propose co-alignment, interaction, and fusion modules to explore the complementary information between the two branches.

3 Real-RawVSR Dataset Construction

Hardware Design. Capturing LR-HR image pairs with short-long focal lengths are common settings for real image SR. This can be easily realized for static scene

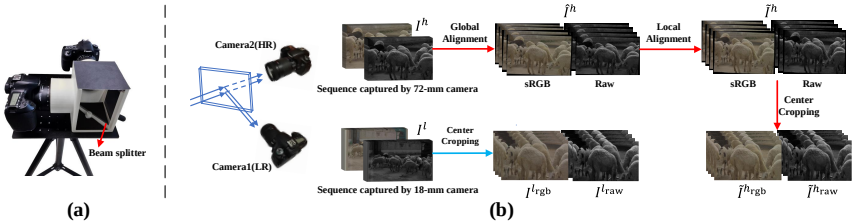

Fig. 1. The capturing hardware (a) and our coarse to fine alignment pipeline to generate aligned LR-HR pairs (b).

by capturing with the same camera [39]. For dynamic LR-HR video capturing, we need to utilize two cameras with different focal lengths. However, this will inevitably bring parallax problems caused by different shooting positions. Inspired by [17,18], which utilizes a beam splitter to divide the incident light into two light beams with a brightness ratio of 1:1, we also utilize this strategy, as shown in Fig. 1 (a). In order to capture LR-HR frame pairs with different ratios, we utilize the DSLR camera with an 18-135mm zoom lens instead of the mobile phone cameras. Therefore, a large beam splitter is expected to cover the lens of DSLR cameras. To this end, we utilize a large and cheap beam splitter with reflectance coating and antireflection coating, instead of a small and expensive beam splitter cube. In order to avoid the influence of natural light from other directions, we design and print a 3D model box to hold the beam splitter. In this way, the two cameras can receive natural light from the same viewpoint. The size of the beam splitter is $150 \times 150 \times 1(\text{mm}^3)$, which is enough to cover the camera lens. We put the camera and beam splitter box on an optical plate, which is installed on a tripod, to improve its stability.

Data Collection. We use two Canon 60D cameras upgraded with a third-party software Magic Lantern [2] to capture raw videos in Magic Lantern Video (MLV) format. To keep the cameras in sync, we use an infrared remote control to signal both cameras to capture at the same time. During capturing, we keep the ISO of the two cameras ranging from 100 to 1600 to avoid noise, and the exposure time ranges from 1/400s to 1/31s to capture both slow and fast motions. All the other settings are set to default values to simulate real capture scenarios. Then we use the MlRawViewer[3] software to process the MLV video to obtain the corresponding sRGB frames and raw frames in the DNG format. For each scene, we capture a short video with six seconds and the frame rate is 25 FPS, namely that each video contains approximately 150 frames in both raw and sRGB formats.

Data Processing. As shown in Fig. 1 (b), although there is no parallax between the LR-HR pair, the field of view (FoV) of the LR frame is much larger than that of the HR frame. In addition, due to the existence of lens distortion, there is still misalignment between the LR-HR pairs. Therefore, we utilize a

[2] https://magiclantern.fm/
[3] https://bitbucket.org/baldand/mlrawviewer/src/master/

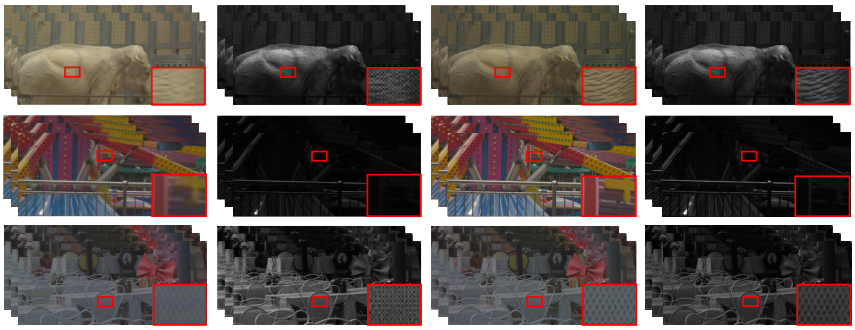

Fig. 2. Examples of videos in Real-RawVSR Dataset with the brightness and contrast of raw frames adjusted for better visualization. From left to right, each column lists LR frames ($I^{l_{\text{rgb}}}$, $I^{l_{\text{raw}}}$) and HR frames ($\tilde{I}^{h_{\text{rgb}}}$, $\tilde{I}^{h_{\text{raw}}}$) in both raw and sRGB domains.

coarse to fine alignment strategy to obtain aligned LR-HR pairs. In the following, we give details for sRGB frame and raw frame alignment, respectively.

1) **RGB frame alignment.** First, we estimate a homography matrix (H) between the upsampled LR ($\hat{I}^{l_{\text{rgb}}}$, the upsampling factor is estimated according to the ratio between the LR and HR focal lengths) and HR ($I^{h_{\text{rgb}}}$) frames using their matched SIFT [25] key points, which are selected by the RANSAC algorithm [16]. Note that, we perform alignment on $I^{h_{\text{rgb}}}$, to make the LR input of our network to be consistent with real captured LR frames, instead of performing alignment on $I^{l_{\text{rgb}}}$ as that in [37]. Then, the aligned HR frame is obtained by $\hat{I}^{h_{\text{rgb}}} = HI^{h_{\text{rgb}}}$. In this way, we can roughly crop the corresponding regions in the LR frame matched with the HR frame. Then, we utilize DeepFlow [34], which is a traditional flow estimation method, to perform pixel-wise alignment for the matching area. Finally, we crop the center area to eliminate the alignment artifacts around the border, generating the aligned LR-HR frames in RGB domain, denoted by ($I^{l_{\text{rgb}}}$, $\tilde{I}^{h_{\text{rgb}}}$).

2) **Raw frame alignment.** The raw frames should go through the same pipeline as that of RGB frames to make $\tilde{I}_t^{h_{\text{raw}}}$ and $\tilde{I}_t^{h_{\text{rgb}}}$ be strictly aligned. However, directly applying the global and local alignment will destroy the Bayer pattern of raw inputs. Therefore, we first pack the Bayer pattern raw frame into RGGB sub-frames, whose size is half of that of RGB frames. Hence, we change the H matrix calculated from sRGB frames by rescaling the translation parameters with a ratio of 0.5. The deep flow vectors are also processed in the same way. In this way, we generate the raw frame pair ($I^{l_{\text{raw}}}$, $\tilde{I}^{h_{\text{raw}}}$). Note that, in this work, we utilize ($I^{l_{\text{raw}}}$, $\tilde{I}^{h_{\text{rgb}}}$) as training pairs. The provided raw pairs can enable future research on raw to raw SR.

We totally captured 600 groups of videos, and manually removed 150 videos with large alignment errors, with 450 videos remaining in our dataset. Fig. 2 gives some examples of our aligned pairs in both raw and sRGB domains. Note that, although they are aligned in spatial, there are still color and illumination differences in each LR-HR pair. These phenomena also exist in other real cap-

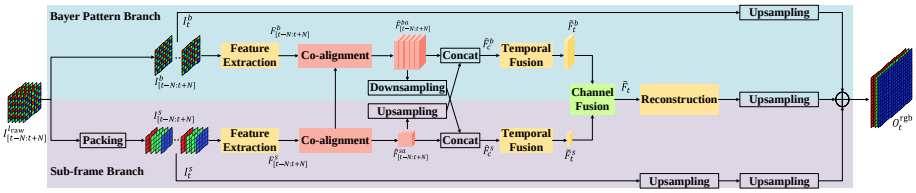

Fig. 3. The proposed Real-RawVSR network for 2× SR. The LR raw sequences ($I^{l_{\text{raw}}}_{[t-N:t+N]}$) are fed into the network in terms of both Bayer pattern and sub-frame forms. The final SR result O^{rgb}_t is obtained by feature interaction and fusion of the Bayer pattern and sub-frame branches.

tured LR-HR pairs [4,37,39]. Our captured scenes vary from indoor to outdoor, and the motion types include camera motions and object motions. The resolution of the original HR frame is 1728×972. After alignment and center cropping, the resolutions of the aligned HR and LR frames for 2× SR are 1440×640 and 720×320, respectively. For each magnification scale, there are 150 video pairs and each video contains about 150 frames. More detailed information about the dataset is presented in the supplementary file.

4 The Proposed Method

We propose a Real-RawVSR network to reconstruct an HR sRGB frame O^{rgb}_t from $2N+1$ consecutive LR raw frames $I^{l_{\text{raw}}}_{[t-N:t+N]}$. The existing raw image (video) SR methods [24,35] usually directly pack the Bayer pattern input into four (RGGB) different channels, where each channel contains the same color pixels. However, this will destroy the pixel order of the original raw frame. Inspired by [22], in this work, we propose to deal with raw frames in two branches, as shown in Fig. 3. The top branch deals with the original Bayer pattern input, and the bottom branch deals with the packed RGGB input. In this way, the top Bayer pattern branch benefits the spatial reconstruction while the bottom sub-frame branch can take advantage of longer neighboring pixels to generate details. To fully take advantage of the complementary information between the two branches, we propose co-alignment, interaction, and fusion modules. In the following, we give details of these modules.

4.1 Packing and Feature Extraction

As shown in Fig. 3, the input LR raw frames $I^{l_{\text{raw}}}_{[t-N:t+N]}$ are fed into the network in different forms for the two branches. The top Bayer pattern branch directly utilizes the raw frames themselves as input. The bottom sub-frame branch utilizes the packed version, namely that we extract the sub-frame with the same color from the Bayer pattern input and all the sub-frames form a new sequence. For simplicity, we denote the input of the Bayer pattern branch as $I^{b}_{[t-N:t+N]}$ and that of the sub-frame branch as $I^{s}_{[t-N:t+N]}$, whose channel number is four

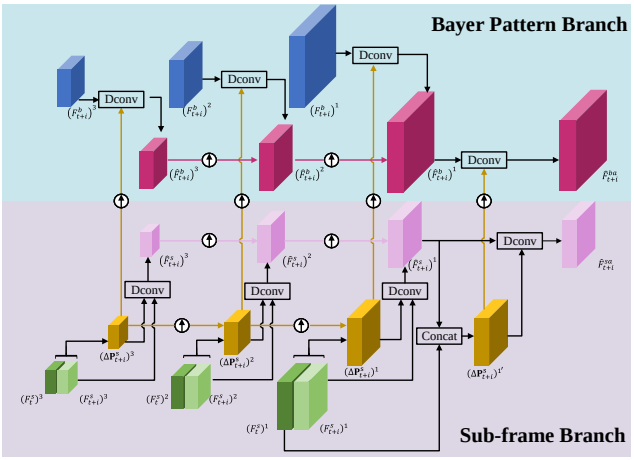

Fig. 4. The proposed co-alignment module. The top branch is for the Bayer pattern feature alignment and the bottom branch is for the sub-frame feature alignment. The two branches share the same offset with different sizes.

times of $I^b_{[t-N:t+N]}$. The Bayer pattern branch keeps the original order of raw pixels, which is good for spatial reconstruction. Although the sub-frame branch cannot keep the original pixel order, it can take advantage of far neighbor correlations to generate details. Therefore, they are complementary to each other, which helps to improve the SR results generated by one single branch. Then, the two inputs go through the feature extraction modules, respectively, where the feature extraction module is constructed by five residual blocks. Note that, the weights for the two feature extraction blocks are not shared since their inputs are in different forms. After the feature extraction module, we obtain $F^b_{[t-N:t+N]}$ with size $(2N+1) \times C \times H \times W$ for the Bayer pattern branch and $F^s_{[t-N:t+N]}$ with size $(2N+1) \times C \times H/2 \times W/2$ for the sub-frame branch, where $2N+1$ is the frame number along the time dimension, C is the channel number, H is the height, and W is the width of features.

4.2 Co-Alignment

Since there are temporal misalignments between neighboring frames, we need to warp neighboring frames to the center frame. Following [32], we utilize PCD alignment. Since we have two branches, a straightforward solution is performing the PCD alignment separately. We note that the two branches actually share the same offset. Therefore, we propose to calculate the alignment offsets from the sub-frame branch and then directly copy the calculated offsets to the Bayer pattern branch to perform the alignment operation. Namely that the two branches are co-aligned.

Given the features of two adjacent frames in the sub-frame branch F^s_t and F^s_{t+i}, we aim to align F^s_{t+i} with F^s_t. The aligned feature $\hat{F}^s_{t+i}$ at position $\mathbf{p}_0$ is

obtained via deformable convolution, which can be expressed by

$$\hat{F}^s_{t+i}(\mathbf{p}_0) = \sum_{k=0}^{K} w_k \cdot F^s_{t+i}(\mathbf{p}_0 + \mathbf{p}_k + \triangle\mathbf{p}_k) \cdot \triangle m_k, \tag{1}$$

where $\mathbf{w}_k$ and $\mathbf{p}_k$ represent the weight and predefined offset for the k-th location in the deformable convolution kernel. The learnable offset $\triangle\mathbf{p}_k$ and the modulation scalar $\triangle m_k$ are predicted from concatenated features of the neighboring and reference frames, denoted by

$$\triangle\mathbf{P}_{t+i} = f([F^s_{t+i}, F^s_t]), \tag{2}$$

where $\triangle\mathbf{P} = \{\triangle\mathbf{p}\}$ represents the set of offsets, and f represents a nonlinear mapping function realized by several convolution layers. For simplicity, we ignore the modulation scalar $\triangle m_k$ in the descriptions and figures. Following PCD alignment, we further utilize pyramidal processing and cascading refinement to deal with large motions, as shown in Fig. 4. The features $(F^s_{t+i})^l$ and $(F^s_t)^l$ are downsampled via strided convolution for $L-1$ times to form a pyramid with L levels. The pyramid features in the Bayer-pattern branch are constructed in the same way. The offsets in the l^{th} level are calculated from the concatenated features in the l^{th} level and the upsampled version of the offsets in the $(l+1)^{\text{th}}$ level. The upsampling is realized by bilinear interpolation and the offset values are magnified by two times. This process is denoted by

$$(\triangle\mathbf{P}^s_{t+i})^l = f([(F^s_{t+i})^l, (F^s_t)^l], 2((\triangle\mathbf{P}^s_{t+i})^{l+1})^{\uparrow 2}). \tag{3}$$

Since the input of the sub-frame branch is actually a down-sampling version of that in the Bayer pattern branch, the offset values for the Bayer pattern branch should be two times of that in the sub-frame branch. Therefore, the offsets for the Bayer pattern branch $(\triangle\mathbf{P}^b_{t+i})^l$ in the l^{th} level can be obtained via two times upsampling and two times magnification of the offsets $(\triangle\mathbf{P}^s_{t+i})^l$ in the sub-frame branch. We denote this process as

$$(\triangle\mathbf{P}^b_{t+i})^l = 2((\triangle\mathbf{P}^s_{t+i})^l)^{\uparrow 2}. \tag{4}$$

Given the offsets, the aligned features for the two branches can be expressed by

$$(\hat{F}^s_{t+i})^l = g(\text{Dconv}((F^s_{t+i})^l, (\triangle\mathbf{P}^s_{t+i})^l), ((\hat{F}^s_{t+i})^{l+1})^{\uparrow 2}), \tag{5}$$

$$(\hat{F}^b_{t+i})^l = g(\text{Dconv}((F^b_{t+i})^l, (\triangle\mathbf{P}^b_{t+i})^l), ((\hat{F}^b_{t+i})^{l+1})^{\uparrow 2}), \tag{6}$$

where g represents the mapping function realized by seveal convolution layers and DConv represents deformable convolution expressed in Eq. 1. Note that, the two-branch DConv shares the same weights in the corresponding level. After alignment for L levels, we further use the offsets $(\triangle\mathbf{P}^s_{t+i})^{1'}$ calculated between $(F^s_t)^1$ and $(\hat{F}^s_{t+i})^1$ to refine $(\hat{F}^s_{t+i})^1$ and $(\hat{F}^b_{t+i})^1$, and generate the final alignment results $\hat{F}^{sa}_{t+i}$ and $\hat{F}^{ba}_{t+i}$ for the neighboring features in the two branches.

We would like to point out that using the proposed co-alignment strategy not only reduces computing complexity but also improves the final SR performance (see the ablation study). The main reason is that the offsets are optimized by both the Bayer pattern features and the sub-frame features, while the offsets calculated with separated alignment can only be optimized with their corresponding features. Therefore, the co-alignment strategy outperforms the sep-alignment.

4.3 Interaction

Since the features in the two branches are complementary, we further propose an interaction module to enrich the feature representations in the two branches. Specifically, the Bayer pattern branch features are downsampled via a 3×3 strided convolution (stride=2) and Leaky Relu layer, and these downsampled features are concatenated with those in the sub-frame branch. Similarly, the sub-frame branch features are upsampled via pixel shuffle [30], which are then concatenated with the features in the Bayer pattern branch. In this way, we generate the interacted features $\hat{F}_c^b \in \mathbb{R}^{(4N+2)\times C\times H\times W}$ and $\hat{F}_c^s \in \mathbb{R}^{(4N+2)\times C\times H/2\times W/2}$.

4.4 Temporal Fusion

Although we have aligned the neighboring frames to the reference frame, these frames still contribute differently to the reference frame SR. Therefore, we utilize attention based fusion to fuse the features together. First, we utilize a non-local temporal attention module [38] to aggregate long-range features to enhance the feature representations along the time dimension. Then, we utilize temporal spatial attention (TSA) [32] based fusion to fuse the features together. Finally, we obtain the temporal fused features $\tilde{F}_t^b$ with size $1 \times C \times H \times W$ and $\tilde{F}_t^s$ with size $1 \times C \times H/2 \times W/2$ for the two branches, respectively.

4.5 Channel Fusion

We utilize channel fusion to merge the features in the two branches together since the same channel of $\tilde{F}_t^b$ and $\tilde{F}_t^s$ may contribute differently to the final SR reconstruction. We adopt selective kernel convolution (SKF) [21] to fuse the two branches via channel-wise weighted average. We first upsample $\tilde{F}_t^s$ via pixel shuffle to make it have the same size as that of $\tilde{F}_t^b$. Then, the two features are added together, going through global average pooling along the channel dimension, generating a channel-wise weighting vector $z \in \mathbb{R}^{1\times 1\times C}$. Then, z goes through the squeeze and excitation layers, generating two weighting coefficients z^b and z^s. Hereafter, they are normalized via softmax, generating the final weighting coefficients $\hat{z}^b$ and $\hat{z}^s$. The final fused feature is obtained by $\tilde{F}_t = \hat{z}^b\tilde{F}_t^b + \hat{z}^s\tilde{F}_t^s$.

4.6 Reconstruction and Upsampling

The fused feature $\tilde{F}_t$ is fed into the reconstruction module, which is realized by 10 ResNet blocks, for the SR reconstruction. After reconstruction, we utilize the

pixel shuffle layer to upsample it and then utilize a convolution layer to generate the three-channel output. We also utilize two long skip connections. One is for the LR Bayer input (I_t^b), which is first processed by a convolution layer and then upsampled by pixel shuffle to a three channel output. The other is for the LR sub-frame input (I_t^s), which is upsampled two times since its spatial size is half of the original input. The three outputs are added together to generate the final HR result O_t^{rgb}. For 4× magnification, similar to EBSR [27], we utilize a two-stage upsampling based long-skip connection.

4.7 Color Correction and Loss Function

As described in Sec. 3, the LR input ($I_t^{l_{\text{rgb}}}$) and ground truth ($\tilde{I}_t^{h_{\text{rgb}}}$) have differences in color and brightness. Directly utilizing pixel-wise loss between the output and the ground truth may lead the network to optimize color and brightness correction other than the essential task of SR, *i.e.,* detail generation. To solve this problem, inspired by [4], we utilize color correction before the loss calculation. Different from [4], we utilize channel-based color correction for RGB channels separately other than calculating a 3×3 color correction matrix to simultaneously correct them. This process can be denoted as

$$\hat{O}_t^c = \alpha^c O_t^c, \alpha^c = \phi(I_t^{l_c}, \tilde{I}_t^{h_c}), c \in \{r, g, b\}, \tag{7}$$

where α^c is the scaling factor for channel c, and it is calculated by minimizing the least square loss between the corresponding pixel pairs in $I_t^{l_c}$ and the downsampled version of $\tilde{I}_t^{h_c}$. Then, we can optimize the network with the Charbonnier loss [20] between the corrected output and the ground truth as $\mathcal{L} = \sqrt{\| \hat{O}_t^{\text{rgb}} - \tilde{I}_t^{h_{\text{rgb}}} \|_2^2 + \epsilon}$, where $\epsilon = 1 \times 10^{-6}$.

5 Experiments

5.1 Training Details

In our experiments, for each magnification factor, 130 videos are used for training and validation, and the other 20 videos are used for testing. To make the movements between neighboring frames more obvious, for each video, we extract frames from the original 150 frames with a step size of three, resulting in a 50-frame sequence. This strategy is also used in [28]. The raw data is pre-processed by black level subtraction and white level normalization. The frame number is 5, *i.e.,* $N = 2$. The channel number C of features is 64. All the convolution filter size is 3×3 [4]. During training, the Bayer pattern patch size is 128×128 and the batch size is 4. We train our model with Adam optimizer and the learning rate is set to 1e-4. The total iteration number is 300k. Our model is implemented in PyTorch and trained with an NVIDIA 3090 GPU.

[4] More details about the network structure are presented in the supplementary file.

Table 1. Quantitative comparison with state-of-the-art VSR methods. The best results are highlighted in bold and the second best results are underlined.

Scale		Bicubic	TOF [36]	TDAN [31]	EDVR [32]	BasicVSR [9]	RawEDVR	DBSR [4]	RawVSR [24]	Ours
2×	PSNR	35.32	35.62	36.14	<u>36.93</u>	36.72	36.74	36.16	36.55	**37.38**
	SSIM	0.9530	0.9555	0.9615	0.9674	0.9668	0.9670	0.9621	<u>0.9677</u>	**0.9705**
3×	PSNR	33.09	33.72	34.43	<u>35.25</u>	34.95	35.23	34.48	34.96	**35.62**
	SSIM	0.9169	0.9241	0.9352	0.9425	0.9408	<u>0.9442</u>	0.9370	0.9431	**0.9468**
4×	PSNR	31.19	32.17	32.84	<u>33.60</u>	33.27	33.55	32.86	33.46	**33.91**
	SSIM	0.8787	0.8928	0.9050	0.9139	0.9113	0.9153	0.9077	<u>0.9164</u>	**0.9182**
Params (M)		-	-	2.3	3.3	6.3	3.3	12.4	4.5	4.8
FLOPs (G)		-	-	360.3	463.3	370.0	464.7	254.7	622.9	494.9

5.2 Comparison with State-of-the-arts

We compare with six state-of-the-art VSR methods, including four methods in sRGB domain (TOFlow [36], EDVR [32], TDAN [31], and BasicVSR [9]) and two methods in raw domain (RawVSR [24] and DBSR [4]). In addition, we also revise EDVR by setting its input to the one channel Bayer pattern input and the original bilinear upsampling operation on the long skip connection is replaced by convolution and pixel shuffle operations. The revised version is denoted as RawEDVR. For a fair comparison, we retrain the above methods on our dataset and add the color correction strategy mentioned in Sec. 4.7 to all the compared methods to avoid the influence of color mis-matching. We use $(I^{l_{\text{rgb}}}_{[t-N:t+N]}, \tilde{I}^{h_{\text{rgb}}}_{t})$ as training pairs for sRGB domain methods and $(I^{l_{\text{raw}}}_{[t-N:t+N]}, \tilde{I}^{h_{\text{rgb}}}_{t})$ for raw domain methods. All the methods are trained with 5 consecutive frames as inputs.

The quantitative comparison results are shown in Table 1. Our method achieves the best results compared to all previous methods on all scaling factors. Specially, for 2× SR, our method outperforms EDVR and RawVSR by 0.45 dB and 0.83 dB, respectively. Note that, although the PSNR results of RawVSR and RawEDVR are worse than those of EDVR, the SSIM results of RawVSR and RawEDVR are generally better than those of EDVR. This demonstrates that the raw input is beneficial for the structure reconstruction, which is also verified by the visual comparison in Fig. 5. We also present the number of parameters and FLOPs (calculated for 4× SR with a 160×360 input) in Table 1. Our method has similar FLOPs as that of RawEDVR and is much lighter than RawVSR. This mainly benefits from the proposed co-alignment strategy, which saves about 100G FLOPs compared with separate alignment.

Fig. 5 presents the visual comparison results for 2× and 4× SR. All the sRGB domain processing methods cannot deal with the false colors embedded in the LR input. RawVSR also cannot remove the false colors since it utilizes the LR sRGB input for guidance. It demonstrates that for real raw VSR, utilizing the LR sRGB input as guidance may be not a good choice. In addition, raw domain processing can generate better details compared with sRGB domain processing

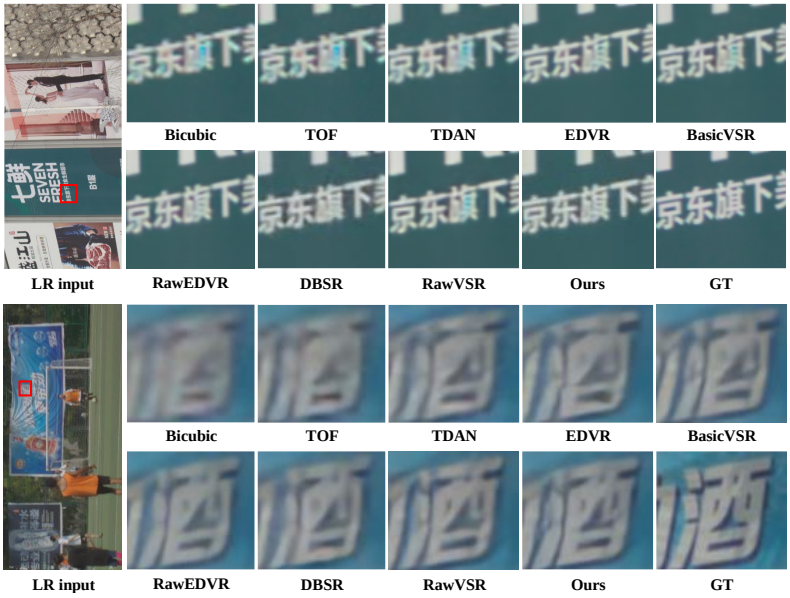

Fig. 5. Visual comparison for 2× and 4× VSR results. For each group, the top (bottom) row presents the results generated by sRGB (raw) domain processing methods.

(see the second image). Our method can correct the false colors well and our generated details are much cleaner than those of other methods.

5.3 Ablation Study

We evaluate the key modules in our network by replacing them with other straightforward solutions. 1) **Co-Alignment.** We evaluate the effectiveness of the co-alignment module by replacing it with a separate alignment, which performs alignment on the two branches separately. As shown in Table 2, co-alignment outperforms sep-alignment by 0.11 dB. The main reason is that the offsets calculated by co-alignment are more accurate than those calculated by sep-alignment. We also present the result by removing the alignment module, which is 0.19 dB less than our proposed method. 2) **Interaction.** If we remove the interaction module from our full model, the result will drop 0.15 dB. It verifies that interaction is beneficial for taking advantage of the complementary information in the two branches. 3) **Channel Fusion.** By replacing the selective kernel based fusion strategy with simple concatenation, the PSNR result will drop 0.06 dB. 4) **Color Correction.** If we do not utilize color correction, the result will be heavily degraded (30.65 dB) due to the color cast. In addition, our channel-based correction is better than the widely used matrix-based correction method by 0.1 dB. 5) **Single Branch.** We also present the results by training the Bayer pattern branch and sub-frame branch separately. We increase the parameters of the two variants by increasing their channel numbers to make

Table 2. Ablation study (4×) for the key modules in our network.

Alignment	Sep-alignment	✗	✓	✗
	Co-alignment	✗	✗	✓
	PSNR/SSIM	33.72/0.9173	33.80/0.9178	**33.91/0.9182**
Interaction	Interaction	✗	✓	
	PSNR/SSIM	33.76/0.9173	**33.91/0.9182**	
Channel Fusion	SKF	✗	✓	
	Concat	✓	✗	
	PSNR/SSIM	33.85/0.9180	**33.91/0.9182**	
Color Correction	Matrix-based	✗	✓	✗
	Channel-based	✗	✗	✓
	PSNR/SSIM	30.65/0.9102	33.81/0.9173	**33.91/0.9182**
Branch	Bayer Branch	✓	✗	✓
	Sub-frame Branch	✗	✓	✓
	PSNR/SSIM	33.73/0.9175	33.70/0.9167	**33.91/0.9182**

their parameters almost the same as that of our full model. Our method outperforms the two variants by nearly 0.2 dB. It demonstrates that the gain of two branch processing is not from the large parameters but from our co-alignment and interaction modules.

6 Conclusion and Discussion

We build the first real-world raw VSR dataset with three magnification ratios in both raw and sRGB domains, which provides a benchmark dataset for both training and evaluation of real raw VSR methods. Based on this dataset, we propose a Real-RawVSR method by dealing with the raw inputs in two branches. By utilizing the proposed co-alignment, interaction, and fusion modules, the complementary information of the two branches is well explored. Experiments demonstrate that the proposed method outperforms state-of-the-art raw and sRGB VSR methods.

Compared with VSR for synthetic LR inputs, dealing with real LR inputs is more difficult due to the color and brightness differences in the LR-HR pair. As reported in [37], the gap between different methods retrained on the same real dataset is much smaller than those trained on the synthetic dataset [32]. In this work, we focus on the network structure design for raw inputs and have achieved impressive gain over our baseline network EDVR. The proposed co-alignment and interaction strategy can be applied to other sRGB VSR methods to improve their performance in dealing with raw inputs. In the future, we would like to explore more effective losses to deal with the color and brightness differences.

References

1. Abdelhamed, A., Afifi, M., Timofte, R., Brown, M.S.: Ntire 2020 challenge on real image denoising: Dataset, methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 496–497 (2020)
2. Abdelhamed, A., Timofte, R., Brown, M.S.: Ntire 2019 challenge on real image denoising: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
3. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 126–135 (2017)
4. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Deep burst super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9209–9218 (2021)
5. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11036–11045 (2019)
6. Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4778–4787 (2017)
7. Cai, J., Gu, S., Timofte, R., Zhang, L.: 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. 0–0 (2019)
8. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: 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)
9. Chan, K.C., Wang, X., Yu, K., Dong, C., Loy, C.C.: 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)
10. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. arXiv preprint arXiv:2104.13371 (2021)
11. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Investigating tradeoffs in real-world video super-resolution. arXiv preprint arXiv:2111.12704 (2021)
12. Chen, C., Xiong, Z., Tian, X., Zha, Z.J., Wu, F.: Camera lens super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1652–1660 (2019)
13. Chen, C., Chen, Q., Do, M.N., Koltun, V.: Seeing motion in the dark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3185–3194 (2019)
14. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3291–3300 (2018)
15. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)

16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
17. Jiang, H., Zheng, Y.: Learning to see moving objects in the dark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7324–7333 (2019)
18. Joze, H.R.V., Zharkov, I., Powell, K., Ringler, C., Liang, L., Roulston, A., Lutz, M., Pradeep, V.: Imagepairs: Realistic super resolution dataset via beam splitter camera rig. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 518–519 (2020)
19. Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K.: Video super-resolution with convolutional neural networks. IEEE transactions on computational imaging **2**(2), 109–122 (2016)
20. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 624–632 (2017)
21. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 510–519 (2019)
22. Liang, C.H., Chen, Y.A., Liu, Y.C., Hsu, W.: Raw image deblurring. IEEE Transactions on Multimedia (2020)
23. Liu, J., Wu, C.H., Wang, Y., Xu, Q., Zhou, Y., Huang, H., Wang, C., Cai, S., Ding, Y., Fan, H., et al.: Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
24. Liu, X., Shi, K., Wang, Z., Chen, J.: Exploit camera raw data for video super-resolution via hidden markov model inference. IEEE Transactions on Image Processing **30**, 2127–2140 (2021)
25. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1150–1157. Ieee (1999)
26. Lugmayr, A., Danelljan, M., Timofte, R.: Ntire 2020 challenge on real-world image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 494–495 (2020)
27. Luo, Z., Yu, L., Mo, X., Li, Y., Jia, L., Fan, H., Sun, J., Liu, S.: Ebsr: Feature enhanced burst super-resolution with deformable alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 471–478 (2021)
28. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
29. Niu, B., Wen, W., Ren, W., Zhang, X., Yang, L., Wang, S., Zhang, K., Cao, X., Shen, H.: Single image super-resolution via a holistic attention network. In: European conference on computer vision. pp. 191–207. Springer (2020)
30. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)

31. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: Tdan: Temporally-deformable alignment network for video super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3360–3369 (2020)
32. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
33. Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., Wang, J.: Practical deep raw image denoising on mobile devices. In: European Conference on Computer Vision. pp. 1–16. Springer (2020)
34. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision. pp. 1385–1392 (2013)
35. Xu, X., Ma, Y., Sun, W.: Towards real scene super-resolution with raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1723–1731 (2019)
36. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. International Journal of Computer Vision **127**(8), 1106–1125 (2019)
37. Yang, X., Xiang, W., Zeng, H., Zhang, L.: 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)
38. Yue, H., Cao, C., Liao, L., Chu, R., Yang, J.: Supervised raw video denoising with a benchmark dataset on dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2301–2310 (2020)
39. Zhang, X., Chen, Q., Ng, R., Koltun, V.: Zoom to learn, learn to zoom. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3762–3770 (2019)
40. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 286–301 (2018)
41. Zhou, K., Li, W., Lu, L., Han, X., Lu, J.: Revisiting temporal alignment for video restoration. arXiv preprint arXiv:2111.15288 (2021)