sensors

MDPI

Article

Lightweight Super-Resolution with Self-Calibrated Convolution for Panoramic Videos

Fanjie Shang [1], Hongying Liu [2,*], Wanhao Ma [1], Yuanyuan Liu [1], Licheng Jiao [1], Fanhua Shang [3], Lijun Wang [4] and Zhenyu Zhou [5]

[1] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[2] The Medical College, Tianjin University, Tianjin 300072, China
[3] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[4] Hangzhou Institute of Technology, Xidian University, Hangzhou 311231, China
[5] Hunan University of Science and Engineering, Yongzhou 425199, China
* Correspondence: hyliu@xidian.edu.cn

Abstract: Panoramic videos are shot by an omnidirectional camera or a collection of cameras, and can display a view in every direction. They can provide viewers with an immersive feeling. The study of super-resolution of panoramic videos has attracted much attention, and many methods have been proposed, especially deep learning-based methods. However, due to complex architectures of all the methods, they always result in a large number of hyperparameters. To address this issue, we propose the first lightweight super-resolution method with self-calibrated convolution for panoramic videos. A new deformable convolution module is designed first, with self-calibration convolution, which can learn more accurate offset and enhance feature alignment. Moreover, we present a new residual dense block for feature reconstruction, which can significantly reduce the parameters while maintaining performance. The performance of the proposed method is compared to those of the state-of-the-art methods, and is verified on the MiG panoramic video dataset.

Keywords: panoramic videos; super-resolution; lightweight network; deformable convolution; self-calibration convolution

Citation: Shang, F.; Liu, H.; Ma, W.; Liu, Y.; Jiao, L.; Shang, F.; Wang, L.; Zhou, Z. Lightweight Super-Resolution with Self-Calibrated Convolution for Panoramic Videos. *Sensors* **2023**, *23*, 392. https://doi.org/10.3390/s23010392

Academic Editors: KWONG Tak Wu Sam, Yun Zhang, Xu Long and Tiesong Zhao

Received: 6 November 2022
Revised: 22 December 2022
Accepted: 26 December 2022
Published: 30 December 2022

1. Introduction

Video super-resolution (VSR) is a classic problem in computer vision, and aims to recover high-resolution videos from low-resolution ones. VSR technology has been widely used in various areas for high-definition displays, such as network videos, digital TV, and surveillance drones. The panoramic video is one class of videos that are real 360-degree omnidirectional sequences, and its pixels are usually arranged in a spherical shape that can provide an immersive experience for users. The panoramic video is the product of a combination of multiple video technologies. Such a video allows the audience to see a wider field of view and a more realistic field of view experience. Because of its 3D stereoscopic characteristics compared with ordinary videos, it is widely used in entertainment, news, the military, and other fields.

In recent years, due to the emergence of deep learning, various methods for video super-resolution based on deep learning have been proposed. For instance, in SOFVSR [1], an optical flow reconstruction network is presented to infer high-resolution (HR) optical flow from coarse to fine, and the motion-compensated low resolution (LR) is input to a super-resolution network to generate the final super-resolved frames. TDAN [2] proposes a temporal deformable network, which utilizes the features of the reference frame and neighboring frames to dynamically predict the offset of the sampling convolutional kernel, and aligns it adaptively at the feature level. EDVR [3] proposes a pyramid, cascade, and deformable convolution (PCD) module. Unlike TDAN, this module performs alignment in

a coarse-to-fine manner and can handle videos with large and complex motions. Moreover, EDVR presents the temporal and spatial attention fusion module, which utilizes temporal attention to concentrate on neighboring frames that are more similar to the reference frame, and uses spatial attention to assign weights to each position in each channel to more effective use of cross-channel and spatial information.

Although the methods mentioned above can achieve good performance for general videos, they may degrade for panoramic videos. Because panoramic videos usually have ultrahigh spatial resolution, they can provide viewers with a strong sense of immersion in the virtual environment. Moreover, the higher the resolution of the camera, the more realistic effect of the panoramic video is. The high resolution requires great hardware performance from camera equipment, and the cost will be largely increased. For this problem, Liu et al. [4] first explored deep learning for super-resolving panoramic videos. Although this method has gained a higher PSNR for panoramic videos, the number of parameters is still high. This issue considerably limits their real-world applications.

In order to balance the performance and the computational cost, we propose a novel lightweight super-resolution framework for panoramic videos. As is known, the alignment between video frames is significant for super-resolution. If more interframe information can be exploited for alignment, it is beneficial to the subsequent reconstruction. Thus, we present a new pooled, self-calibrated convolution (PSCC) for frame alignment, which significantly reduces the complexity of the deformable convolution and achieves accurate alignment in a gradual manner. Moreover, in the reconstruction operation, we design a new lightweight residual dense block to further reduce the complexity of the model. Our method achieves a balance between algorithm performance and complexity.

The main contributions of this work are listed as follows.

- We propose the first lightweight panoramic video super-resolution (LWPVSR) method for panoramic video super-resolution, which can achieve a good balance between performance and complexity. To the best of our knowledge, this is the first proposition of a lightweight panoramic VSR framework.
- Moreover, we present a new pooled, self-calibrated convolution for frame alignment. The self-calibrated convolution is introduced to make the learned offset more accurate in a progressive manner and reduce the complexity of the proposed network.
- Finally, we design a new significantly lighter residual dense block (LWRDB) for feature reconstruction, which achieves the purpose of reducing the complexity of the model while maintaining the performance of our method. Many experimental results verify the advantage of the proposed LWPVSR method against state-of-the-art methods.

The rest of this paper is organized as follows. Some related works on super-resolution of panoramic videos are introduced in Section 2. Section 3 describes the proposed lightweight super-resolution method in detail. In Section 4, we demonstrate the experimental results of our method. Finally, we show the conclusions and future work in Section 5.

2. Related Work

2.1. Super-Resolution Methods for Ordinary Videos

Most of video super-resolution methods (e.g., VESPCN [5], TDAN [2], SOFVSR20 [6], and EDVR [3]) have been proposed to address ordinary videos. They have improved the performance of restored high-resolution videos. For example, Yi et al. [7] proposed a general omniscient framework to leverage the LR framework and estimated hidden states from the past, present, and future frames. Benefiting from the global information feature of OVSR [7], the OVSR method refreshes the metrics on the Vid4 test set.

2.2. Super-Resolution Methods for Panoramic Videos

There are many image super-resolution methods such as [8–13]. For instance, ref. [11] can utilize the plenoptic geometry of the scene to perform alignment between consecutive frames in a video sequence and employ all visual information to generate high-resolution panoramic images. In [10], the spherical Fourier transform (SFT) was calculated based

on the nonuniform sampling data on the sphere, which can transform low-resolution panoramic images with arbitrary rotation to reconstruct a high-resolution panoramic image. The joint alignment and super-resolution problem is converted into a least square minimization problem in the SFT domain. In [14], the authors introduced SRCNN [15], which is the earlier work to use deep learning for super-resolution of panoramic images. It fine tuned the SRCNN by optimizing input size and using the panoramic training set to adapt the fine-tuned method to the features of the high-resolution panoramic images. Based on the existing viewport-based panoramic image transmission system, ref. [16] proposed a framework that used the high-resolution content of the viewport to improve the quality of the surrounding low-resolution areas. The adaptive initial viewport of each image was predicted in view of contextual similarity of the sphere, so as to provide more useful information for low-resolution regions.

Only a few works involve the super-resolution of panoramic videos. As we have mentioned in Section 1, Liu et al. [4] designed a single frame and multiframe joint network for the super-resolution of panoramic videos, which explored both the spatial information and the temporal information. In addition, deformable convolutions are employed to eliminate the motion difference between feature maps of the target frame and its neighboring frames. Although it achieves sound results, the network contains a large number of parameters, resulting in low computational efficiency, which is not unfavorable for the promotion of practical applications. Therefore, this paper will propose the first lightweight video super-resolution method for panoramic videos.

3. The Proposed Lightweight Architecture for Panoramic Video Super-Resolution

In this section, we propose the first lightweight panoramic video super-resolution (LWPVSR) method. The proposed LWPVSR method mainly consists of the four main modules: the feature extraction module, the feature alignment module, the reconstruction module, and the dual network module, as shown below.

3.1. Our Network Architecture

As shown in Figure 1, the network structure of our method mainly consists of three main parts, which are the feature extraction module, the feature alignment module, and the reconstruction module. The backbone network learns the residual images of the video frames, then sums them with the direct upsampling results of the target frames to obtain the final super-resolution results. In addition, super-resolution is an ill-posed problem—that is, mapping a low-resolution video to high-resolution is a one-to-many problem. In order to reduce the solution space of the super-resolution, we introduce a dual mechanism to the backbone network, and it learns a dual regression mapping, which can increase the constraints on LR videos—that is, the duality mechanism acts as a subsupervised network to enhance the performance of SR. The whole super-resolution process of the proposed method is expressed as follows,

$$\tilde{I}_t = H_{LWPVSR}(\hat{I}_{t-N:t+N}), \tag{1}$$

where $H_{LWPVSR}(\cdot)$ denotes the proposed algorithm network, N is the temporal radius (e.g., $N = 3$), and $\tilde{I}_t$ and $\hat{I}_t$ are the super-resolution result of the target frame and the low resolution of the target frame, respectively.

3.2. The Feature Extraction Module

The feature extraction module is responsible for extracting the features of the input video frames to prepare for subsequent feature alignment. In the proposed method, the feature extraction module is composed of several residual blocks, which mainly consist of two convolutional layers. The residual block is more conducive to training. Therefore, the number of parameters is very small, and it maintains network performance in combina-

tion with other modules. The process of the proposed feature extraction module can be formulated as follows:

$$F = H_{FE}(\hat{I}_{t-N:t+N}). \quad (2)$$

Note that $H_{FE}(\cdot)$ denotes the feature extraction operation, and F denotes the extracted features.

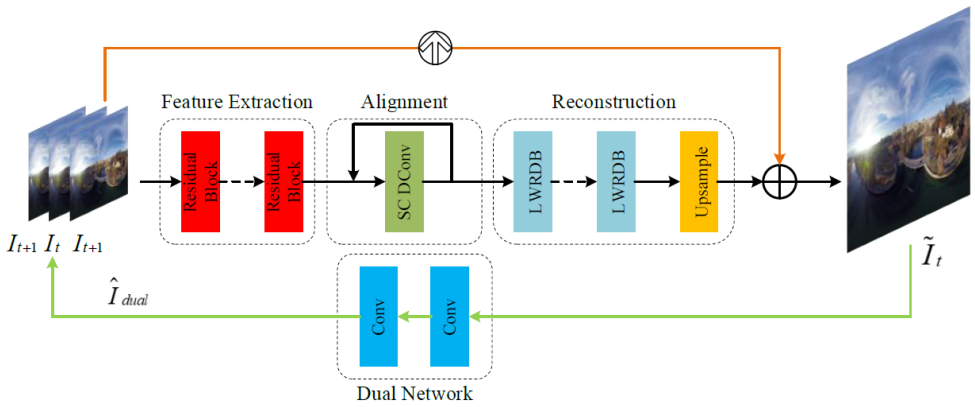

Figure 1. The network architecture of the proposed LWPVSR method. Our LWPVSR method mainly consists of the four modules: the feature extraction module, the feature alignment module, the reconstruction module, and the dual network module.

3.3. The Proposed Feature Alignment Module

In this subsection, we propose a new module for feature alignment between frames based on deformable convolution. As is known, the effectiveness of deformable convolution in video super-resolution has been witnessed and confirmed in EDVR [3]. In EDVR, the deformable convolution was integrated into a pyramid, cascading, and deformable convolution (PCD) module in EDVR, as shown in Figure 2. In fact, PCD has a pyramid-like structure. The top layer is a lower-resolution feature map, and the bottom layer is for the reference frame and neighboring frames. Different layers represent the feature information of different frequencies. PCD first aligns the reference feature map with the smallest resolution to form a rough alignment, and then transfers the offset and aligned feature map to a layer with a larger resolution, so that the offset and continuously aligned feature map are passed to the bottom layer every time. Thus, an implicit motion compensation from coarse to fine is formed from top to bottom. However, PCD introduced a large number of parameters and computational cost. In order to reduce the parameters, here we design a new pooled, self-calibrated convolution (PSCC) to replace the pyramid cascading structure and maintain the multi-scale learning capability, as shown in Figure 3.

Inspired by the self-calibrated convolution in [17], our PSCC module employs a pooling for downsampling to expand the receptive field, so as to learn more contextual information without increasing the network complexity. Specifically, after the target features and the neighboring features are merged, a convolution operation is performed, and then through channel splitting, the channel is divided into two, one channel only performs a simple convolution. In the other channel, we utilize a new upsample operation to make the learned features match with the scale. The learned features via the Sigmoid activation are again multiplied with the features through channel splitting. Finally, the features from the two channels are concatenated for output. Our PSCC module is embedded in the deformable convolution to learn the information about the neighbors of the reference frame, rather than the global information, so as to avoid the pollution information of other irrelevant frames and achieve more accurate frame alignment. From the perspective of the structures of PCD and PSCC, PCD uses multiple deformable convolutional networks (DCNs) and convolutional networks to cascade and form a pyramid structure, and each DCN is based on a feature map of different levels. Although our PSCC only adopts one deformable convolutional network, and combines it with self-correcting convolution, which

not only reduces the number of parameters, but also learns more accurate offsets to achieve better alignment. This will be verified by the subsequent experimental results. Compared with the PCD module with 1.38 M parameters in EDVR, the number of the parameters of our PSCC module is only 0.04 M.

The process of the feature alignment in our LWPVSR method is expressed as follows,

$$F^a_{t\pm i} = H_{Alignment}(F_t, F_{t\pm i}), \tag{3}$$

where $H_{Alignment}(\cdot)$ denotes our alignment operation and F_t and $F_{t\pm i}$ denote the features of the target frame and the nearest neighboring frame, respectively. $F^a_{t\pm i}$ denotes the aligned features of each frame. Here, we use F^a to represent the result of all the aligned frames.

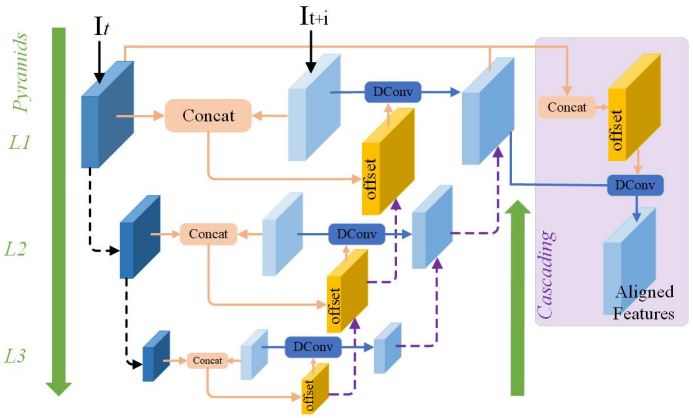

Figure 2. The structure of the PCD module in EDVR [3].

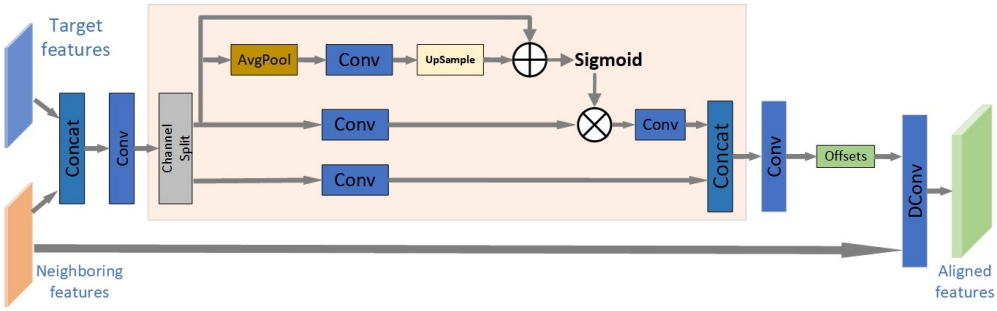

Figure 3. Our proposed pooled self-calibrated convolution (PSCC) module for feature alignment.

3.4. Our Reconstruction Module

In the proposed reconstruction module, inspired by the residual dense blocks (RDB) in [18], as shown in Figure 4a, a lightweight residual dense block (LWRDB) is designed for feature transformation and restoration. Its detailed structure is shown in Figure 4b.

First, the input features F_{in} through a layer of convolution output the corresponding feature maps. The channel is then shuffled by a channel shuffle operation, followed by a channel split operation which divides the number of channels into two proportionally. One part is fed into the convolution, and the other is connected to the output of the convolution in a jump connection. The channel shuffle and channel split operations are executed again, and so on. Finally, the number of channels is reduced through a 1×1 convolution, and the output is added to the initial input of the module to obtain the final result F_{out} of the module. The process is given by

$$F_{out} = H_{LWRDB}(F_{in}), \tag{4}$$

where $H_{LWRDB}(\cdot)$ denotes the operation of the LWRDB module. It is noted that in our design, the introduction of the channel shuffle [19] and the channel split [20] is important

compared with that of Figure 4a. The purpose of the channel shuffle operation is to break up the output of the previous layer of convolution in channel dimension, and the purpose of the channel split operation is to split the channel into two according to a presetting ratio, as shown in Figure 4b. The purpose of combining these two operations is to reduce the number of parameters while still making full use of all levels of features like residual dense blocks, so that the number of parameters can be reduced. Meanwhile, it maintains high performance. It should be noted that we have verified in our experiment that the number of parameters of the RDB module is 1.08 M, while that of ours is only 0.81 M. Obviously, the number of parameters of our proposed LWRDB is smaller. In fact, our lightweight residual block in the reconstruction module introduces both channel shuffle and channel split operations. The channel shuffle can strengthen the exchange of information between channels, and channel split can reduce the number of parameters. Compared with the reconstruction module in Figure 4a, our method has fewer parameters and can maintain the performance of the network.

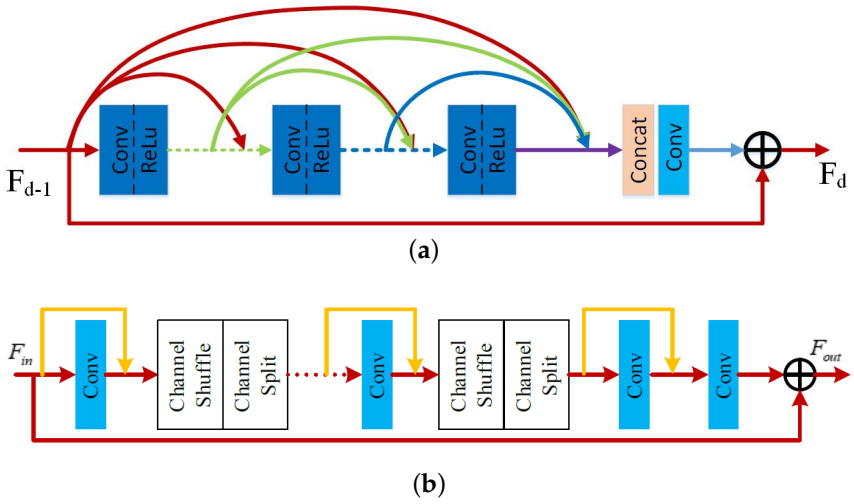

Figure 4. Comparison of the structures of the residual dense block (RDB) used in [19,20] and our lightweight RDB. (**a**) Existing RDB [19,20]. (**b**) Our lightweight RDB.

3.5. Our Dual Network Module and Loss Function

In our proposed method, the final super-resolution result is obtained by adding the output of the reconstruction module to the result of the upsampled target frame. It is expressed as follows,

$$\tilde{I}_t = H_{reconst}(F^a) + \hat{I}_t \uparrow, \tag{5}$$

where $H_{reconst}(\cdot)$ denotes the mapping function of reconstruction module. Here $\uparrow$ denotes upsampling, and $\tilde{I}_t$ denotes the super-resolution result of the target frame.

In order to show the important content in the equatorial region in the panorama video, we introduce a weighted mean square error (WMSE) loss, which is defined as follows,

$$\frac{1}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\omega(i,j)}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\omega(i,j)\cdot(\tilde{I}_t(i,j)-I_t(i,j))^2, \tag{6}$$

where M and N denote the width and height of one frame, respectively, (i,j) represents the coordinate position of each pixel in a frame, and $\omega(i,j)$ is the weight at the corresponding pixel position, which is allocated according to the pixel position and is given by

$$\omega(i,j) = \cos\frac{(j+0.5-\frac{N}{2})\pi}{N}, \tag{7}$$

where $i = i_0, i_0 + 1, \cdots, i_0 + wd - 1$ and $j = j_0, j_0 + 1, \cdots, j_0 + h - 1$. Here, (i_0, j_0) represents the upper left corner of the patch, wd is the width of the patch, and h is the height of the patch.

In order to explain the weight change in each frame more intuitively, we show it visually in Figure 5. The black and white color represent the distribution of weights. The lighter the color, the greater the weight is, and the darker the color, the smaller the weight is. That is, the weights gradually decrease from the equator to the two polar regions. The weights are assigned on the whole frame during data processing.

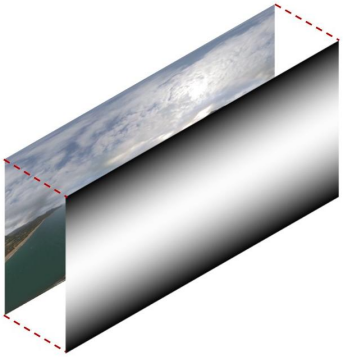

Figure 5. The weight diagram of the loss function.

The loss function in our architecture is composed of two parts. One is from the main branch—$L_{primary}$ (i.e., input, feature extraction, alignment, reconstruction, and output)—and the other is from the dual subnetwork: L_{dual}. The overall loss function of the proposed method is formulated as follows,

$$L_{total} = L_{primary} + \lambda L_{dual}, \tag{8}$$

where $L_{primary}$ and L_{dual} are both calculated by Equation (6). The parameter λ is a balance factor between $L_{primary}$ and L_{dual}.

It is noted that compared with ordinary videos, the information of panoramic video is distributed on a sphere instead of a plane. The panoramic video, which is essentially a spherical video, cannot directly use the storage structure and encoding algorithm designed for ordinary videos. The current mainstream solution is to use the mapping relationship to project the spherical video onto the plane and compress the obtained plane video, and the equirectangular projection (ERP) is widely used. In this case, the important content is usually displayed in the equatorial region, and the less content is at the poles. Moreover, because the information of the panoramic video is distributed in a spherical shape, the features in the same dimensionality are more uneven, and the video is more prone to be distorted. Addressing the particularity that more content distributed at the equator and less content at the poles, we used the weighted loss function, as shown in Equation (6). It aims to increase the weight of the equatorial region and reduce the weight of the polar region. Addressing the features distributed on the same dimension are more uneven or the offset is too large, we think that using deformable convolution is not sufficient to solve this issue. Therefore, we propose to adopt self-correcting convolution combined with deformable convolution—that is, our PSCC module to learn these offset features. It is more conducive to achieving better alignment results.

4. Experimental Results

In this section, we compare the proposed LWPVSR method with eight state-of-the-art super-resolution algorithms for panoramic video super-resolution tasks.

4.1. Datasets

The MiG dataset [4] is utilized for evaluating the performance of super-resolution of the proposed LWPVSR method. The data set has 200 videos for training and eight videos for test. We adopt the bicubic interpolation algorithm to 2× downsample each video frame as the ground truth (GT). Then, we further perform 4× downsampling on GT to obtain the corresponding LR video. Moreover, in order to demonstrate the superiority of the proposed LWPVSR method, we also collected another video sequence from the Internet, named Clip_009, and adopted it for performance evaluation.

The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are usually used as indicators to measure the performance of all the video super-resolution algorithms. In order to make a fair comparison, similar to other works, all indices are calculated on the Y channel for all the algorithms. Different from ordinary videos, we also use the two video quality metrics (i.e., WS-PSNR and WS-SSIM) in [4] to measure the performance of all the methods.

4.2. Training Setting

We implemented all the models in the PyTorch framework and used two NVIDIA Titan XP GPUs for training. The training schemes and parameters of other methods are listed below.

(1) SR360 [8]: The batch size is set to 16. The weights of all the layers were initialized randomly and the network was trained from the scratch. The network used the Adam solver with a learning rate, 1×10^{-4}.
(2) VSRnet [21]: The batch size is 240, a learning rate of is 1×10^{-4} used for the first two layers, 1×10^{-5} for the last layer and a weight decay rate of 0.0005 are set as in [21].
(3) FRVSR [22]: The Adam is an optimizer. The learning rate is fixed at 1×10^{-4}. Each sample in the batch is a set of 10 consecutive video frames, i.e., 40 video frames are passed through the networks in each iteration.
(4) VESPVN [5]: The initial batch size is 1. Every 10 epochs the batch size is doubled until it reaches a maximum size of 128. The optimizer is Adam with a learning rate, 1×10^{-4}.
(5) TDAN [2]: The batch size is set to 64. The Adam is the optimizer. The learning rate is initialized to 1×10^{-4} for all layers and decreases half for every 100 epochs.
(6) SOFVSR [6]: The batch size is 32. The optimizer is Adam. The initial learning rate is 1×10^{-3} and divided by 10 after every 80 K iterations.
(7) EDVR [3]: The batch size is set to 32. The learning rate is initialized to 4×10^{-4}, and initializes deeper networks by parameters from shallower ones for faster convergence.
(8) OVSR [7]: The batch size is 16. The optimizer is Adam. The initial learning rate is 1×10^{-3} and decays linearly to 1×10^{-4} after 120 K iterations, which keeps the same until 200 K iterations. Then the learning rate is further decayed to 5×10^{-5} and 1×10^{-5} until convergence.

In our method, the feature extraction module is composed of three residual blocks, each residual block consists of two layers of convolution, and the number of channels is set to 64. The reconstruction module includes five LWRDB blocks, each block is composed of six convolutional layers, and the number of channels is 64. In our experiment, we convert the video frames from the RGB space to the YCbCr space and then use the Y channel as the input to our network. Unless stated otherwise, the network takes three consecutive video frames as inputs. The input patch size is 64 × 64, and the batch size is set to 32. Moreover, we also employ data enhancement techniques as in other methods, including reflection, random cropping, and rotation. Furthermore, we defined the ratio for channel split by experience. If the ratio is larger than 0.5, it means that more features do not participate in the subsequent calculations but are directly cascaded to the subsequent feature maps. Then the following convolutional layers will be meaningless. If the ratio is smaller than 0.5, the model parameters will increase and it results in a higher computational cost. Therefore,

the ratio equaling to 0.5 is a balanced choice. During training, we optimize the network by using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is set to 2×10^{-4}, and then is reduced by half after every 20 epochs. In our loss function, through experiments and experiences, the value of the parameter λ is set to 0.1. And the performance of each method has been optimized with its hyperparameter tuning to show their best results in our experiments.

4.3. Quantitative Comparison

We also implemented nine other state-of-the-art VSR algorithms for performance comparison. They include bicubic, SR360 [8], VSRnet [21], VESPCN [5], FRVSR [22], TDAN [2], SOFVSR20 [6], EDVR [3], and OVSR [7]. The quantitative results including PSNR/WS-PSNR, SSIM/WS-SSIM, inference time, and floating point operations per second (FLOPs) of all the methods on representative video clips are shown in Tables 1 and 2, respectively.

Table 1. Comparison of all the methods in terms of PSNR (top) and SSIM (bottom).

	Bicubic	**SR360 [8]**	**VSRnet [21]**	**FRVSR [22]**	**VESPCN [5]**	**TDAN [2]**	**SOFVSR20 [6]**	**EDVR [3]**	**OVSR [7]**	**LWPVSR**
Clip_005	26.38 0.6868	26.58 0.7101	26.59 0.7075	25.36 0.7095	26.71 0.7203	26.73 0.7213	26.72 0.7203	26.73 0.7217	26.69 0.7207	**26.81** **0.7251**
Clip_006	30.09 0.8494	30.69 0.8580	30.39 0.8573	29.70 0.8700	31.04 0.8723	31.11 0.8740	31.16 0.8775	31.48 0.8685	29.72 0.8902	**31.58** **0.8902**
Clip_007	27.65 0.8119	29.29 0.8406	28.10 0.8245	28.90 0.8458	29.50 0.8490	29.61 0.8534	29.54 0.8527	30.18 0.8630	29.31 0.8622	**30.99** **0.8700**
Clip_008	31.88 0.9005	32.22 0.9001	32.15 0.9069	31.83 0.9162	32.63 0.9134	32.74 0.9150	32.70 0.9147	32.91 0.9186	32.81 0.9183	**33.02** **0.9183**
Average	29.00 0.8121	29.69 0.8272	29.30 0.8241	28.95 0.8353	29.97 0.8388	30.05 0.8409	30.03 0.8413	30.32 0.8479	29.97 0.8457	**30.60** **0.8507**
Params. (M)	-	0.58	0.16	5.05	0.86	1.96	1.05	20.60	3.48	2.30
Time (ms)	-	64.30	2.52	71.57	122.59	16.11	76.86	670.80	69.55	92.31
FLOPs (T)	-	0.457	0.018	0.348	0.007	0.558	0.135	0.954	0.201	0.204

Table 2. Comparison of all the methods in terms of WS-PSNR (top) and WS-SSIM (bottom).

	Bicubic	**SR360 [8]**	**VSRnet [21]**	**FRVSR [22]**	**VESPCN [5]**	**TDAN [2]**	**SOFVSR20 [6]**	**EDVR [3]**	**OVSR [7]**	**LWPVSR**
Clip_005	26.39 0.6888	26.62 0.7131	26.60 0.7118	25.37 0.7257	26.75 0.7263	26.78 0.7267	26.77 0.7257	26.80 0.7293	26.73 0.7260	**26.84** **0.7298**
Clip_006	28.64 0.8274	29.37 0.8422	28.94 0.8386	28.27 0.8574	29.63 0.8569	29.71 0.8594	29.74 0.8622	30.04 0.8744	29.72 0.8685	**30.15** **0.8759**
Clip_007	29.75 0.8009	30.76 0.8214	30.15 0.8165	30.23 0.8374	31.24 0.8379	31.42 0.8406	31.29 0.8392	31.57 0.8464	31.55 0.8465	**31.86** **0.8482**
Clip_008	30.46 0.8685	30.85 0.8726	30.72 0.8779	30.34 0.8869	31.19 0.8854	31.28 0.8885	31.24 0.8880	31.43 0.8929	31.34 0.8924	**31.52** **0.8938**
Average	28.81 0.7964	29.40 0.8123	29.10 0.8112	28.55 0.8268	29.70 0.8266	29.80 0.8288	29.76 0.8288	29.96 0.8358	29.84 0.8333	**30.09** **0.8369**
Params. (M)	-	0.58	0.16	5.05	0.86	1.96	1.05	20.60	3.48	2.30
Time (ms)	-	64.30	2.52	71.57	122.59	16.11	76.86	670.80	69.55	92.31
FLOPs (T)	-	0.457	0.018	0.348	0.007	0.558	0.135	0.954	0.201	0.204

It can be seen that our LWPVSR method obtains the highest PSNR and SSIM results, and the amount of its parameters is relatively small. Our LWPVSR method performs much better than EDVR in terms of PSNR/WS-PSNR and SSIM/WS-SSIM, and the the former has significantly fewer parameters than the latter (i.e., 2.30 M vs. 20.60 M). That is, LWPVSR is nearly 1/10 size of EDVR. It is because the proposed PSCC module in our LWPVSR plays an important role, and decreases the PCD module in EDVR by man parameters but maintains the performance. In addition, compared with FRVSR, our model parameters are 2.7 M smaller, and the PSNR of model is 1.65 dB higher than FRVSR. SR360, VSRnet, VESPCN, TDAN, and SOFVSR20 are relatively lightweight video super-resolution architectures, with

model parameters below 2.0 M. However, the performance of all of them is significantly lower than that of the proposed method. Moreover, the PSNR and WS-PSNR results of all the methods on other video clips are shown in Tables 3 and 4. We can see that the results of our LWPVSR method are much better than those of the state-of-the-art methods. All the experimental results show that our LWPVSR method can achieve a good balance between the model complexity and performance.

Table 3. Comparison of all the methods in terms of PSNR (top) and SSIM (bottom) on other video clips.

	Bicubic	**SR360 [8]**	**VSRnet [21]**	**FRVSR [22]**	**VESPCN [5]**	**TDAN [2]**	**SOFVSR20 [6]**	**EDVR [3]**	**OVSR [7]**	**LWPVSR**
Clip_001	27.57 0.8659	29.06 0.8833	27.75 0.8742	28.80 0.8920	28.56 0.8909	29.20 0.8965	29.08 0.9004	29.44 0.9108	29.25 0.9122	**29.68** **0.9176**
Clip_002	26.06 0.7426	27.26 0.7866	26.54 0.7650	27.42 0.8045	27.20 0.7976	27.43 0.8052	27.39 0.8073	27.58 0.8138	27.87 0.8378	**27.75** **0.8231**
Clip_003	25.68 0.8240	26.45 0.8495	25.95 0.8359	26.39 0.8551	26.52 0.8568	26.63 0.8623	26.55 0.8607	26.66 0.8663	26.57 0.8737	**26.82** **0.8700**
Clip_004	30.61 0.8889	31.46 0.8931	31.08 0.8983	32.25 0.9220	32.17 0.9196	32.44 0.9257	32.46 0.9280	33.03 0.9379	32.72 0.9404	**33.66** **0.9412**
Clip_009	26.03 0.7515	27.23 0.7957	26.50 0.7717	27.40 0.8123	27.16 0.8044	27.40 0.8131	27.36 0.8136	27.56 0.8224	27.79 0.8637	**29.41** **0.8801**
Average	27.19 0.8146	28.29 0.8416	27.56 0.8290	28.45 0.8572	28.32 0.8539	28.62 0.8606	28.57 0.8620	28.85 0.8702	28.84 0.8856	**29.46** **0.8864**
Params. (M)	-	0.58	0.16	5.05	0.86	1.96	1.05	20.60	3.48	2.30
Time (ms)	-	64.30	2.52	71.57	122.59	16.11	76.86	670.80	69.55	92.31
FLOPs (T)	-	0.457	0.018	0.348	0.007	0.558	0.135	0.954	0.201	0.204

Table 4. Comparison of all the methods in terms of WS-PSNR and WS-SSIM on other video clips.

	Bicubic	**SR360 [8]**	**VSRnet [21]**	**FRVSR [22]**	**VESPCN [5]**	**TDAN [2]**	**SOFVSR20 [6]**	**EDVR [3]**	**OVSR [7]**	**LWPVSR**
Clip_001	29.84 0.9630	30.86 0.8771	30.19 0.8731	30.95 0.8909	31.12 0.8901	31.35 0.8948	31.35 0.8969	31.89 0.9082	31.67 0.9053	**32.04** **0.9071**
Clip_002	25.81 0.7416	27.02 0.7792	26.27 0.7626	27.12 0.7975	26.89 0.7916	27.12 0.7978	27.03 0.8003	27.32 0.8082	27.59 0.8226	**27.45** **0.8112**
Clip_003	24.49 0.7807	25.17 0.8134	24.75 0.7972	25.12 0.8197	25.23 0.8212	25.33 0.8275	25.26 0.8252	25.37 0.8339	25.26 0.8308	**25.48** **0.8312**
Clip_004	29.88 0.8666	30.87 0.8802	30.39 0.8796	31.66 0.9077	31.59 0.9053	31.91 0.9122	31.90 0.9143	32.45 0.9249	32.18 0.9242	**32.89** **0.9263**
Clip_009	25.78 0.7448	26.99 0.7815	26.24 0.7617	27.09 0.7971	26.87 0.7907	27.09 0.7972	27.03 0.7993	27.29 0.8076	27.98 0.8312	**28.11** **0.8387**
Average	27.16 0.7993	28.18 0.8263	27.57 0.8148	28.39 0.8426	28.34 0.8398	28.56 0.8459	28.51 0.8472	28.86 0.8566	28.94 0.8628	**29.20** **0.8629**
Params. (M)	-	0.58	0.16	5.05	0.86	1.96	1.05	20.60	3.48	2.30
Time (ms)	-	64.30	2.52	71.57	122.59	16.11	76.86	670.80	69.55	92.31
FLOPs (T)	-	0.457	0.018	0.348	0.007	0.558	0.135	0.954	0.201	0.204

In order to demonstrate the relation between performance and parameters more clearly, the visualized diagram is also shown in Figure 6. It can be seen that our method attains a higher performance at the cost of lower numbers of parameters.

4.4. Qualitative Comparison

In this subsection, we qualitatively compare our method with the other methods on video sequences Clip_001, Clip_003, Clip_004 and Clip_009, as shown in Figures 7–10, respectively.

It can be seen that our LWPVSR method has achieved much better performance than other methods, including EDVR with 20.60 M parameters, and they have superior visual results in all these figures. For example, in Figure 7, the image recovered by our LWPVSR method seems more real, which is closer to the original high-resolution image. However, the images recovered by other methods, such as TDAN, FRVSR, and SOFVSR20, are blurry.

Similar results can also be observed from Figures 8–10. In general, compared with other methods, our LWPVSR method achieves a better balance between the model complexity and algorithm performance, resulting in less distortion and more reliable results in the panoramic video super-resolution task.

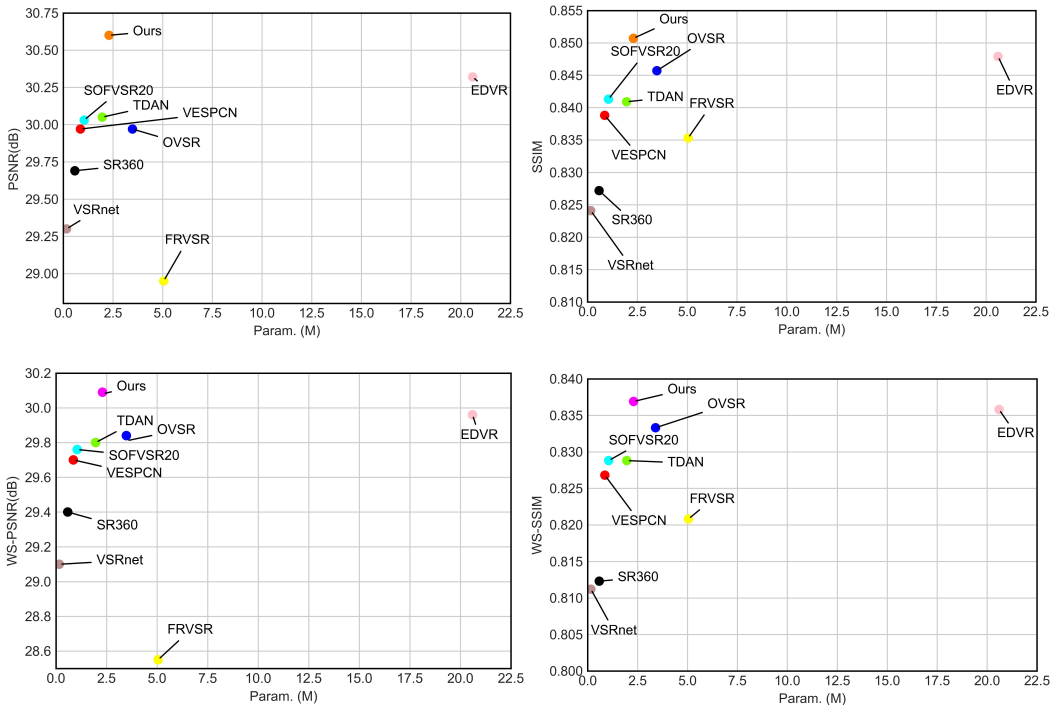

Figure 6. Comparison of all the methods in terms of performance and number of parameters. Note that the *y*-axis represents different performance metrics (including PSNR, SSIM, WS-PSNR, and WS-SSIM), and the *x*-axis corresponds to the number of parameters in different methods.

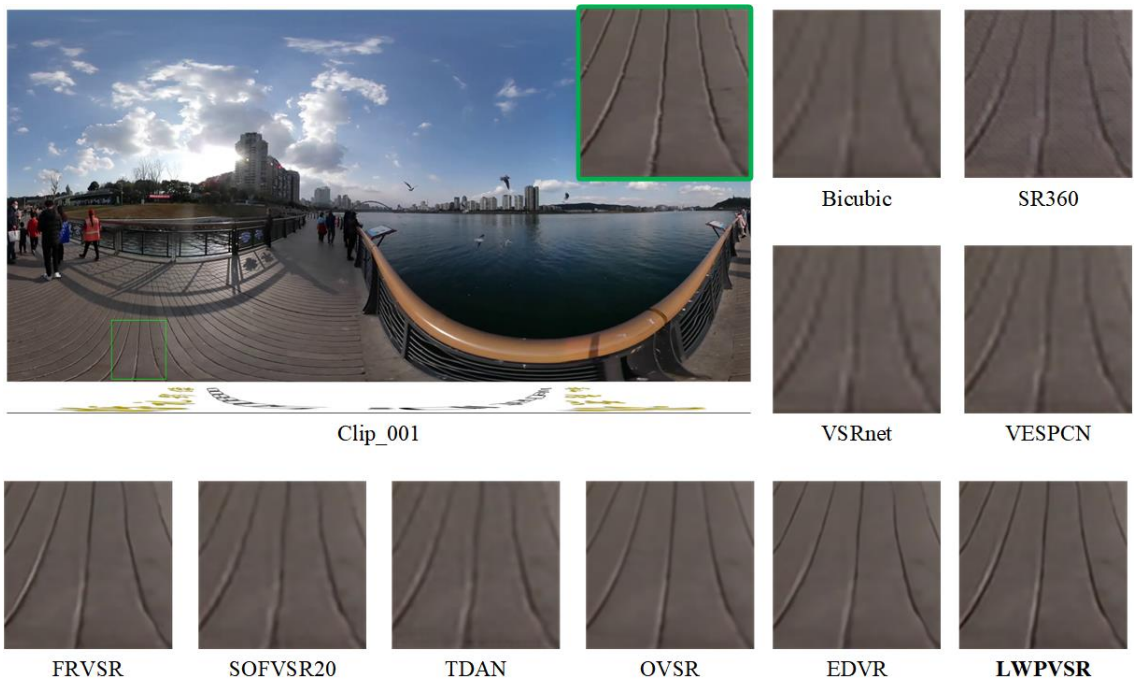

Figure 7. The results of all the algorithms performing 4× super-resolution on Clip_001 of the MiG test set.

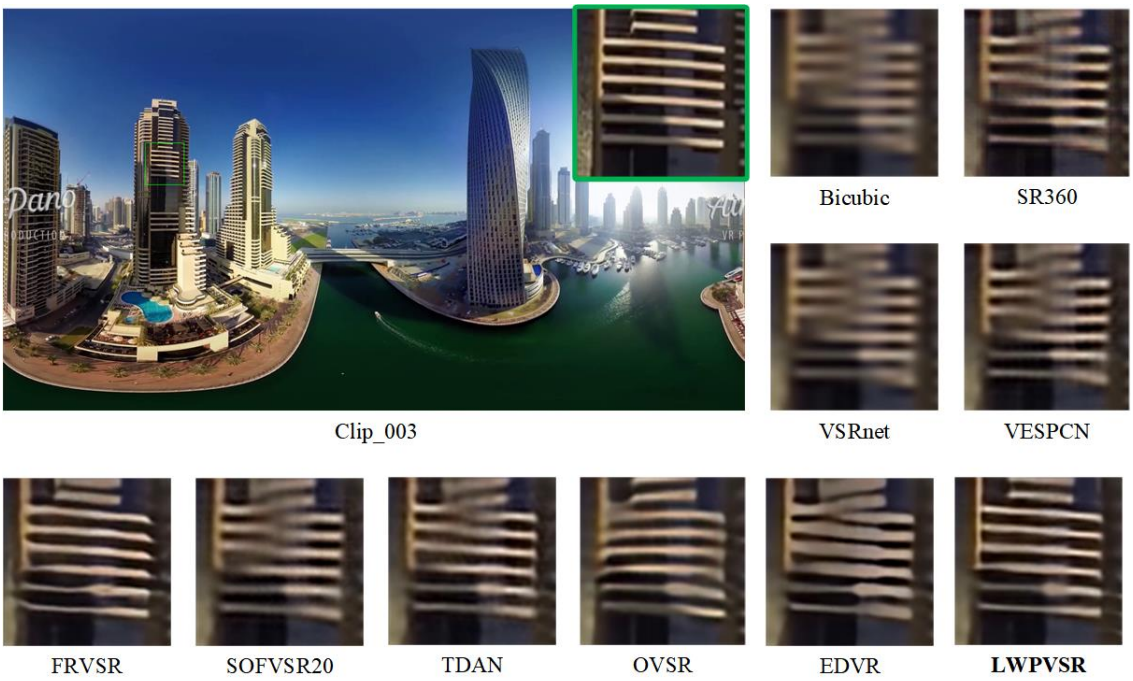

Figure 8. The results of all the algorithms performing 4× super-resolution on Clip_003 of the MiG test set.

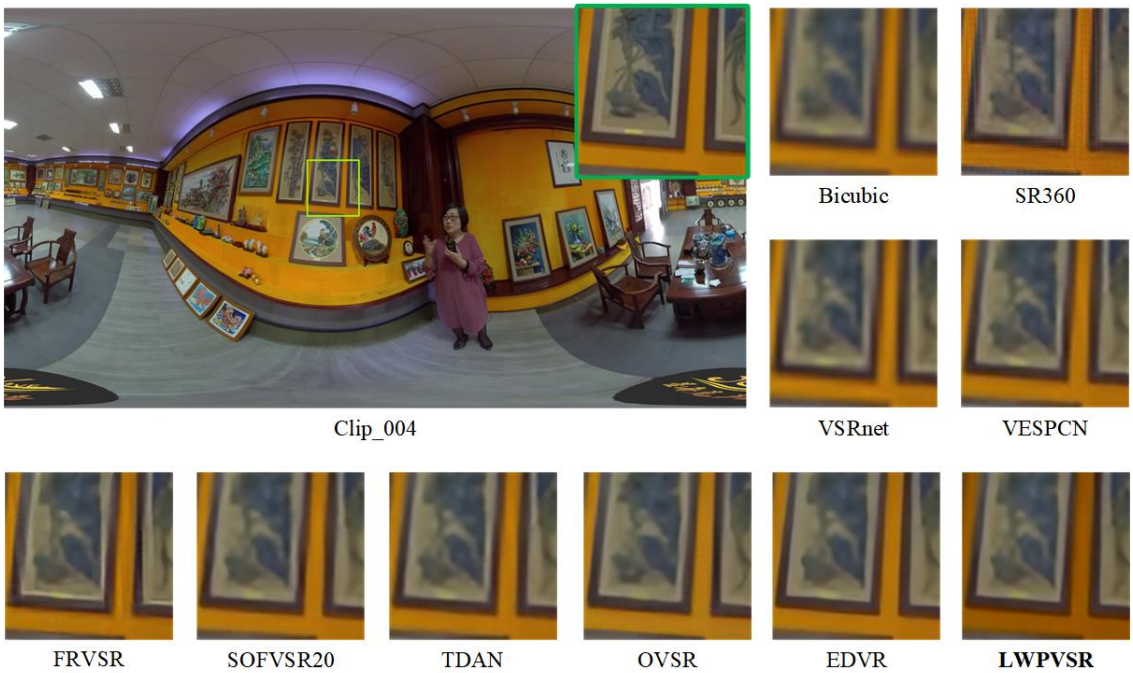

Figure 9. The results of all the algorithms performing 4× super-resolution on Clip_004 of the MiG test set.

Figure 10. The results of all the algorithms performing 4× super-resolution on Clip_009 of the MiG test set.

4.5. Ablation Studies

In this subsection, we analyze the contribution of each module in our network, mainly including PSCC and LWRDB, as shown in Table 5. The baseline is our architecture, as shown in Figure 1. The PSNR and SSIM results are 30.60 dB and 0.8507, respectively. When the architecture is without the PSCC module, the PSNR drops by 0.30 dB, and the number of parameters decreases 0.04 M. The performance drops by 0.92 dB when the baseline is without the LWRDB module. Moreover, without PSCC and LWRDB, the PSNR result decreases by 0.96 dB. All the results also verify the importance of the proposed modules, including PSCC and LWRDB for the proposed method.

Table 5. Ablation studies for each module in the proposed LWRDB network.

	PSNR	**SSIM**	**Parameters (M)**
Ours	30.60	0.8507	2.30
Ours *w*/*o* PSCC	30.30	0.8471	2.26
Ours *w*/*o* LWRDB	29.68	0.8290	1.49
Ours *w*/*o* PSCC and LWRDB	29.64	0.8285	1.44

5. Conclusions and Future Work

In this paper, a lightweight and efficient panoramic video super-resolution method was designed from the perspective of lightweight networks. This method adopts deformable convolution to align the nearest neighbor features with the target feature, in order to further enhance the alignment effect step. In particular, we introduced self-calibrated convolution to gradually implement the alignment operation in a recursive manner. Moreover, we also proposed a lighter and more efficient LWRDB module based on the RDB module. Various experimental results verified the effectiveness of the proposed method. Compared with mainstream video super-resolution algorithms, our proposed method achieves a better balance between performance and algorithm complexity.

In the future, we will design more effective strategies, such as the attention strategy [23] for the lightweight architecture to further enhance the performance while maintaining its cost.

Author Contributions: Methodology, F.S. (Fanjie Shang) and H.L.; Validation, Z.Z.; Formal analysis, Y.L., F.S. (Fanhua Shang) and Z.Z.; Data curation, W.M.; Project administration, L.W.; Funding acquisition, L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 61976164, 62276182 and 61876221), and Natural Science Basic Research Program of Shaanxi (Program No. 2022GY-061).

Acknowledgments: We thank all the reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, L.; Guo, Y.; Lin, Z.; Deng, X.; An, W. Learning for video super-resolution through HR optical flow estimation. In Proceedings of the Asian Conference on Computer Vision (ACCV), Perth, Australia, 2–6 December 2018; pp. 514–529.
2. Tian, Y.; Zhang, Y.; Fu, Y.; Xu, C. TDAN: Temporally-deformable alignment network for video super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3360–3369.
3. Wang, X.; Chan, K.C.K.; Yu, K.; Dong, C.; Loy, C.C. EDVR: Video restoration with enhanced deformable convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019; pp. 1954–1963.
4. Liu, H.; Ruan, Z.; Fang, C.; Zhao, P.; Shang, F.; Liu, Y.; Wang, L. A single frame and multi-frame joint network for 360-degree panorama video super-resolution. *arXiv* **2020**, arXiv:2008.10320.
5. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4778–4787.
6. Wang, L.; Guo, Y.; Liu, L.; Lin, Z.; Deng, X.; An, W. Deep video super-resolution using HR optical flow estimation. *IEEE Trans. Image Process.* **2020**, *29*, 4323–4336. [CrossRef] [PubMed]
7. Yi, P.; Wang, Z.; Jiang, K.; Jiang, J.; Lu, T.; Tian, X.; Ma, J. Omniscient video super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021; pp. 4409–4418.
8. Ozcinar, C.; Rana, A.; Smolic, A. Super-resolution of omnidirectional images using adversarial learning. In Proceedings of the 21st International Workshop on Multimedia Signal Processing (MMSP), Kuala Lumpur, Malaysia, 27–29 September 2019; pp. 1–6.
9. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.
10. Arican, Z.; Frossard, P. Joint registration and super-resolution with omnidirectional images. *IEEE Trans. Image Process.* **2011**, *20*, 3151–3162. [CrossRef] [PubMed]
11. Bagnato, L.; Boursier, Y.; Frossard, P.; Vandergheynst, P. Plenoptic based super-resolution for omnidirectional image sequences. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Hong Kong, China, 26–29 September 2010; pp. 2829–2832.
12. Rivadeneira, R.E.; Sappa, A.D.; Vintimilla, B.X.; Hammoud, R. A Novel Domain Transfer-Based Approach for Unsupervised Thermal Image Super-Resolution. *Sensors* **2022**, *12*, 2254. [CrossRef] [PubMed]
13. Kim, B.; Jin, Y.; Lee, J.; Kim, S. High-Efficiency Super-Resolution FMCW Radar Algorithm Based on FFT Estimation. *Sensors* **2021**, *21*, 4018. [CrossRef] [PubMed]
14. Fakour-Sevom, V.; Guldogan, E.; Kämäräinen, J.-K. 360 panorama super-resolution using deep convolutional networks. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Funchal, Portugal, 27–29 January 2018; Volume 1.
15. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 184–199.
16. Li, S.; Lin, C.; Liao, K.; Zhao, Y.; Zhang, X. Panoramic image quality-enhancement by fusing neural textures of the adaptive initial viewport. In Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Atlanta, GA, USA, 22–26 March 2020; pp. 816–817.
17. Liu, J-J.; Hou, Q.; Cheng, Mi.; Wang, C.; Feng, J. Improving convolutional networks with self-calibrated convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10093–10102.
18. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
20. Ma, N.; Zhang, X.; Zheng, Ha.; Sun, J. Shufflenet V2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 122–138.

21. Kappeler, A.; Yoo, S.; Dai, Q.; Katsaggelos, A.K. Video super-resolution with convolutional neural networks. *IEEE Trans. Comput. Imaging* **2016**, *2*, 109–122. [CrossRef]
22. Sajjadi, M.S.M.; Vemulapalli, R.; Brown, M. Frame-recurrent video super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6626–6634.
23. Du, J.; Cheng, K.; Yu, Y.; Wang, D.; Zhou, H. Panchromatic Image super-resolution via self attention-augmented wasserstein generative adversarial network. *Sensors* **2021**, *21*, 2158. [CrossRef] [PubMed]