More and Larger Auxiliary Feature-Guided Spatial-Temporal Super-Resolution for Rendered Sequences

Lijie Zheng and Xiao Liang(✉)

School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China
xiaoliang.edu@foxmail.com

Abstract. The post-processing rendered sequences improves the quality of the sequences and shortens the time of the rendering phase. However, most of the current post-processing methods for sequences are suitable for video. Directly transferring these methods to rendered sequences cannot obtain high-quality results. To address this problem, we propose an end-to-end spatial-temporal super-resolution network for rendered sequences, which improves rendering efficiency by simultaneously implementing frame interpolation (FI) and super-resolution. In the FI task, accurately inferring results that closer to the real motion state of the target frames is the key and difficult point to ensure the generation effect. For this issue, we design an auxiliary feature-guided interpolation (AFGI) module. By introducing the auxiliary features corresponding to the target frames, AFGI module provides the real motion state of the target frames to the network. In the part of aggregating contextual information, we propose a weighted aggregation upsampling (WAUpS) module. Aggregation is selectively performed based on the correlation between incoming information and the current frame. WAUpS module effectively prevents irrelevant information from affecting the super-resolution results, which is a problem with the direct aggregation methods used previously. At the same time, WAUpS module combines the upsampled features with the corresponding high-resolution auxiliary features. This addition provides the output with rich detail textures and other key information, further improving the overall processing quality. Experimental results show that compared with state-of-the-art (SOTA) methods, our method not only obtains high-quality rendered sequences processing results, but also effectively improves the rendering efficiency.

Keywords: Spatial-temporal super-resolution · Auxiliary features · Rendered image sequences.

1 Introduction

Photo-realistic rendering algorithms have a wide and diverse application for generating rendered sequences in advertising design, video games, and computer

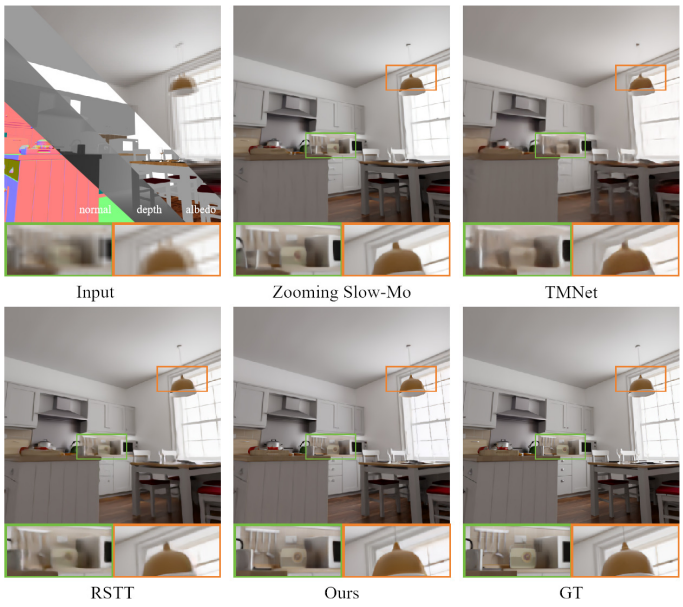

Fig. 1. Compared with the three SOTA STVSR methods, our network can achieve more accurate results in processing rendered sequences.

animations. Taking Monte Carlo (MC) path tracing algorithm for example [1], during the rendering process, numerous intricate calculations, such as ray reflections and refractions per pixel, are required to achieve high-quality rendering results. However, this consumes a substantial amount of time, often requiring dozens of hours to generate rendered sequences. Due to the inherent computational complexity of rendering algorithms, it is quite challenging to adapt to the continuous evolution of display devices and meeting the ongoing demand for high-quality rendered animation. Therefore, enhancing the efficiency of rendering sequences remains a significant challenge.

Recently, many efforts have been devoted to accelerating rendering. The core idea is to employ post-processing strategies and design denoising or super-resolution neural network [2–5], which shifts heavy computation burden from rendering pipeline to more efficient deep-learning mapping translation. Nonetheless, those methods mainly focus on improving rendering efficiency of single-frame image rather than sequences. On the other hand, in the community of video process, significant progress have been made for improving video resolution and frame rates. The video super-resolution (VSR) networks [6, 7] reconstruct low-resolution (LR) video into corresponding high-resolution (HR) video. The video frame interpolation (VFI) networks [8, 9] use the existing set of frames to synthesize the additional subset of frames, thereby increasing the frame rate of the video. The space–time video super-resolution (STVSR) networks [10, 11]

consider the commonalities of super-resolution and interpolation, and integrate these two tasks into one network. However, although these networks can effectively process video, they cannot produce high-quality processing results when applied to rendered sequences. Therefore, in order to improve rendering efficiency more effectively, it is necessary to design a post-processing method that is more suitable for rendered sequences.

Unlike video recorded by electronic devices, the generation of rendered sequences is often accompanied by the by-product of auxiliary features. Since the auxiliary features only need to consider the basic features of the object surface, they can be quickly rendered. Moreover, because auxiliary features contain rich information such as structure and texture, they are widely used in post-processing methods such as rendered image denoising [12, 13] and real-time rendering supersampling [14]. Many of these post-processing methods only use the auxiliary features corresponding to the input image, resulting in limited information they can provide. Therefore, when designing post-processing methods, it is a worthwhile way to use richer auxiliary features to help process rendered content.

Combining the above problems, we propose an end-to-end spatial-temporal super-resolution (STSR) network that is more suitable for rendered sequences. The network takes the high-frame-rate (HFR) and HR auxiliary feature sequences corresponding to the target sequences as one of the inputs, providing richer additional information for processing low-frame-rate (LFR) and LR input rendered sequences. Among them, the auxiliary features are albedo, depth and normal. To improve the quality of the processing results, we design two important modules: auxiliary feature-guided interpolation (AFGI) and weighted aggregation upsampling (WAUpS). The AFGI module uses auxiliary features unique to the rendered sequences to make up for the problem of inaccurate estimation of the motion state of intermediate frames in previous FI methods. When calculating the intermediate offsets between the target intermediate frames and the adjacent input frames, auxiliary features corresponding to the target frames and the input frames are introduced. These auxiliary features provide the real motion state of the target frames, which helps to calculate the more accurate offsets, thereby improving the accuracy of the target frames. The WAUpS module improves the traditional multi-frame alignment method by adding a similarity calculation component. It effectively reduces the impact of irrelevant information on super-resolution results by selectively aggregating contextual information based on the similarity between the incoming information and the current frame. In addition, the WAUpS module also upsamples the aggregated features so that they can be fused with the corresponding HR auxiliary features, providing clearer details, textures and other information for the final super-resolution results.

In summary, the contributions of this paper are as follows: 1) We propose an end-to-end network for rendered image sequences, which reconstructs low-quality rendered sequences into high-quality ones by jointly implementing interpolation and super-resolution, thereby effectively improving rendering efficiency. 2) By introducing HFR and HR auxiliary feature sequences corresponding to the tar-

get sequences, the network is provided with more accurate structure and texture information, helping to improve the quality of the processing results. 3) We design the AFGI module and the WAUpS module to more fully integrate auxiliary features into the joint task and improve the processing efficiency of the network.

2 Related Work

2.1 Video Frame Interpolation

Most of traditional VFI methods are based on optical flow, which uses the input images to approximate the intermediate optical flow and thus generate the corresponding intermediate frames. For example, Super SloMo [15] linearly fuses the bidirectional optical flow between the input images to approximate the intermediate optical flow. DAIN [16] designs a depth-aware flow projection layer to generate the intermediate optical flow. RRIN [17] proposes a residual refinement technique to calculate the optical flow and generate intermediate frames. These methods all use the bidirectional optical flow between input frames to indirectly calculate the intermediate optical flow. To improve the efficiency of the algorithm, people propose methods to directly estimate the intermediate optical flow. RIFE [8] directly uses the input adjacent images to calculate the intermediate optical flow, and introduces privileged distillation to guide the generation of more accurate intermediate flow. In addition, since flow-based methods have high requirements on the accuracy of optical flow estimation, some unrelated flow methods are proposed. AdaCoF [18] synthesizes intermediate frames by estimating the kernel weight and offset vector of each target pixel. To further improve the accuracy of generating intermediate frames, DSepConv [19] is proposed to adaptively estimate kernels, offsets, and masks from less but more relevant pixel information.

2.2 Video Super-Resolution

Different from single image super-resolution, VSR methods often improve reconstruction effects by aligning multiple frames. RBPN [20] adopts the idea of back projection [21, 22] in multi-image super-resolution, and circularly integrates the spatial and temporal contexts from consecutive frames. BasicVSR [6] improves the propagation and alignment parts of the traditional VSR methods, and proposes the cyclic structure of bidirectional flow and the feature alignment method based on optical flow. The framework has high expressiveness and scalability. For example, IconVSR and BasicVSR++ [23] further optimize the propagation and alignment parts based on it to achieve better algorithm performance. With the continuous development of deformable convolution [24, 25], TDAN [26] proposes to use deformable alignment to adaptively align reference frames and each support frame at the feature level. EDVR [27] uses a pyramid structure on the basis of TDAN to align frame features in a coarse-to-fine manner, further improving the advantages of deformable alignment in super-resolution.

2.3 Space-Time Video Super-Resolution

The STVSR methods integrate VSR and VFI into one framework to improve the frame rate and resolution of the video at the same time. STARnet [28] makes full use of the mutual information relationship between time and space, higher resolution provides richer motion cues, and higher frame rate helps better pixel alignment. RSTT [11] combines interpolation and super-resolution into one module, which reduces the complexity of the model and accelerates the processing speed. However, these methods only use short-term features to synthesize intermediate frames, and cannot make full use of the information of continuous input frames. Therefore, STDAN [29] uses the deformable attention network to explicitly utilize temporal context information to assist interpolation and super-resolution reconstruction. Zooming Slow-Mo [10] proposes to use the deformable ConvLSTM [30] to simultaneously align and aggregate time information. On the basis of Zooming Slow-Mo, TMNet [31] adds a temporal modulation block to implement interpolation of arbitrary intermediate frames. LTFA-Net [32] designs a long-term mixture of experts module to extract complementary spatial-temporal information from multiple consecutive adjacent frame features and use it for features interpolation.

3 Network Architecture

3.1 Framework Overview

The architecture of our network is shown in Fig. 2, which mainly includes four parts: feature extraction module, auxiliary feature-guided interpolation (AFGI) module, weighted aggregation upsampling (WAUpS) module and reconstruction module.

The feature extraction module extracts the initial feature sequence $\{F_{2t-1}^{LR}\}_{t=1}^{N}$ from LR rendered sequence $\{I_{2t-1}^{LR}\}_{t=1}^{N}$ using one convolution layer and five residual blocks, and converts HR auxiliary feature sequence $\{G_t^{HR}\}_{t=1}^{2N-1}$ into $\{A_t^{HR}\}_{t=1}^{2N-1}$ using an encoder network consisting of five convolution layers (the first four with LeakyReLU activation function). Then, the AFGI module (Section 3.2) generates the intermediate frame feature between each two adjacent frame features, and the corresponding auxiliary features are introduced to help more accurately distort to the intermediate moment, so as to obtain the more accurate HFR feature sequence $\{F_t^{LR}\}_{t=1}^{2N-1}$. In order to improve the clarity of the final processing result, the WAUpS module (Section 3.3) is used to aggregate more relevant contextual information for the HFR feature sequence and auxiliary feature sequence through bidirectional propagation, and corresponding auxiliary features are added to the HFR feature sequence after upsampling. By reconstructing the final processed feature sequence $\{F_t^{HR}\}_{t=1}^{2N-1}$, high-quality HFR and HR image sequence $\{I_t^{HR}\}_{t=1}^{2N-1}$ can be obtained.

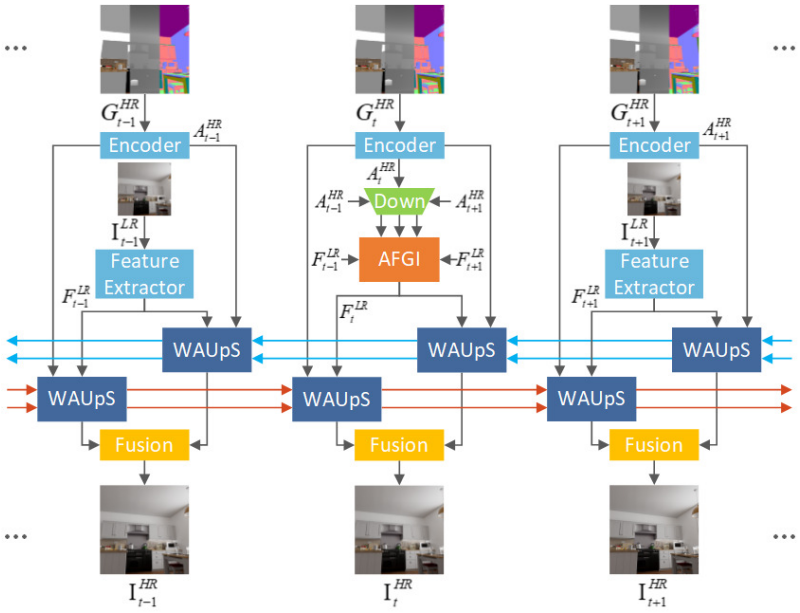

Fig. 2. Overview of our STSR Network for Rendered Sequences. Note that the red and blue arrows indicate the forward and backward propagation of features through the entire sequence, respectively.

3.2 Auxiliary Feature-Guided Interpolation (AFGI)

Most of the previous FI methods use existing frames to estimate and simulate the motion state of the intermediate frames, resulting in a certain gap between the final generated intermediate frames and the real intermediate frames. To fill the gap of the networks for the motion state of the intermediate frames, we use the unique advantages of the rendered sequences to introduce the auxiliary features corresponding to the intermediate frames to help generate the intermediate frames more accurately. Fig. 3 shows the process of generating intermediate frame feature F_t^{LR} between adjacent input frame features F_{t-1}^{LR} and F_{t+1}^{LR} using the AFGI module.

Before introducing auxiliary features, the bidirectional offsets $offset_{t-1\to t+1}$ and $offset_{t+1\to t-1}$ are calculated using F_{t-1}^{LR} and F_{t+1}^{LR}, and then linearly fused [15] to obtain the estimated intermediate offsets $offset'_{t-1\to t}$ and $offset'_{t+1\to t}$ from the intermediate time (T=0.5) to the adjacent frames.

$$\begin{aligned} offset_{t-1\to t+1} &= Conv(F_{t-1}^{LR}, F_{t+1}^{LR}) \\ offset_{t+1\to t-1} &= Conv(F_{t+1}^{LR}, F_{t-1}^{LR}) \end{aligned} \tag{1}$$

$$\begin{aligned} offset'_{t-1\to t} &= -(1-T)T{\cdot}offset_{t-1\to t+1} + T^2{\cdot}offset_{t+1\to t-1} \\ offset'_{t+1\to t} &= (1-T)^2{\cdot}offset_{t-1\to t+1} - T(1-T){\cdot}offset_{t+1\to t-1} \end{aligned} \tag{2}$$

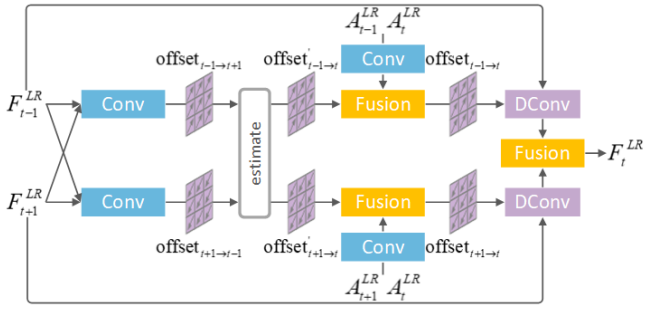

Fig. 3. Architecture of the AFGI module.

Afterwards, corresponding LR auxiliary features A^{LR} are incorporated to improve the accuracy of estimated intermediate offsets and obtain the intermediate offsets $offset_{t-1\to t}$ and $offset_{t+1\to t}$. The LR auxiliary features A^{LR} are downsampled through stride convolution filter, enabling them to be fused with the estimated intermediate offsets.

$$\begin{aligned} offset_{t-1\to t} &= fusion(offset'_{t-1\to t}, Conv(A^{LR}_{t-1}, A^{LR}_{t})) \\ offset_{t+1\to t} &= fusion(offset'_{t+1\to t}, Conv(A^{LR}_{t+1}, A^{LR}_{t})) \end{aligned} \tag{3}$$

Based on the outstanding contribution of deformable convolution in images alignment, we use it to align adjacent frame features to generate alignment features $F^{LR}_{t-1\to t}$ and $F^{LR}_{t+1\to t}$ at intermediate moment. Then, the alignment features are fused to obtain intermediate frame feature F^{LR}_{t}.

$$\begin{aligned} F^{LR}_{t-1\to t} &= DConv(F^{LR}_{t-1}, offset_{t-1\to t}) \\ F^{LR}_{t+1\to t} &= DConv(F^{LR}_{t+1}, offset_{t+1\to t}) \end{aligned} \tag{4}$$

$$F^{LR}_{t} = fusion(F^{LR}_{t-1\to t}, F^{LR}_{t+1\to t}) \tag{5}$$

By performing the above operation on every two adjacent frames in the entire LFR feature sequence $\{F^{LR}_{2t-1}\}^{N}_{t=1}$, the HFR feature sequence $\{F^{LR}_{t}\}^{2N-1}_{t=1}$ can be obtained.

3.3 Weighted Aggregation Upsampling (WAUpS)

Recently, aligning supporting frames to reference frames to aggregate more contextual information is an essential step in VSR. However, this direct aggregation method can also introduce some irrelevant or less relevant information, resulting in poor processing results. To ensure that the aggregated contextual information is advantageous, we proposes the WAUpS module (Fig. 4) that aggregates contextual information based on weights. Meanwhile, similar to the idea of introducing auxiliary features in the AFGI module, this module also introduces HR auxiliary features to provide more additional information for features upsampling and improve the quality of processing results.

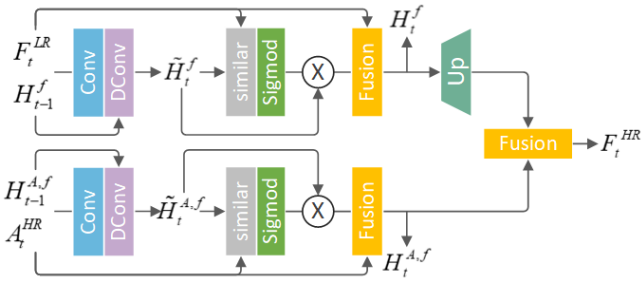

Fig. 4. Architecture of the WAUpS module.

Because the resolution of features and auxiliary features is different, the same weighted aggregation is performed on them respectively. Take the forward processing of the tth feature F_t^{LR} (or auxiliary feature A_t^{HR}) as an example. Firstly, the hidden state H_{t-1}^f (or $H_{t-1}^{A,f}$) from the $t-1$th feature is aligned to F_t^{LR} (or A_t^{HR}) to obtain the initial hidden state $\tilde{H}_t^f$ (or $\tilde{H}_t^{A,f}$) of the current frame. Then, the similarity between F_t^{LR} and $\tilde{H}_t^f$ (or A_t^{HR} and $\tilde{H}_t^{A,f}$) is calculated and normalized by the Sigmod activation function to obtain the fusion weight W_t (or W_t^A).

$$\begin{aligned} \tilde{H}_t^f &= DConv(H_{t-1}^f, Conv(H_{t-1}^f, F_t^{LR})) \\ \tilde{H}_t^{A,f} &= DConv(H_{t-1}^{A,f}, Conv(H_{t-1}^{A,f}, A_t^{HR})) \end{aligned} \tag{6}$$

$$\begin{aligned} W_t &= Sigmod(similar(F_t^{LR}, \tilde{H}_t^f)) \\ W_t^A &= Sigmod(similar(A_t^{HR}, \tilde{H}_t^{A,f})) \end{aligned} \tag{7}$$

Finally, $\tilde{H}_t^f$ (or $\tilde{H}_t^{A,f}$) is weighted and fused with F_t^{LR} (or A_t^{HR}) to obtain the final hidden state H_t^f (or $H_t^{A,f}$). The PixelShuffle [33] is used to upsample H_t^f, so that it can be fused with $H_t^{A,f}$ to obtain the tth forward propagation HR feature $F_t^{HR,f}$.

$$\begin{aligned} H_t^f &= fusion((W_t \cdot \tilde{H}_t^f), F_t^{LR}) \\ H_t^{A,f} &= fusion((W_t^A \cdot \tilde{H}_t^{A,f}), H_t^{HR}) \end{aligned} \tag{8}$$

$$F_t^{HR,f} = fusion(Up(H_t^f), H_t^{A,f}) \tag{9}$$

To ensure that each feature aggregates the same amount of contextual information, two WAUpS operations are performed on each feature using bidirectional propagation to obtain backward and forward propagation HR features (such as $F_t^{HR,b}$, $F_t^{HR,f}$), and the corresponding HR features are fused to obtain the final HR feature (such as F_t^{HR}).

$$F_t^{HR} = fusion(F_t^{HR,b}, F_t^{HR,f}) \tag{10}$$

Table 1. The input content of our network.

Type	Frames	Resolution		
		Training	Validation	Testing
Rendered images	4	$64 * 64$	$128 * 128$	$128 * 128$
Albedo	7	$256 * 256$	$512 * 512$	$512 * 512$
Depth	7	$256 * 256$	$512 * 512$	$512 * 512$
Normal	7	$256 * 256$	$512 * 512$	$512 * 512$

3.4 Loss Function

To better measure the performance of the network and guide the update direction and step size of the network parameters, we use the Charbonnier [34] function as the reconstruction loss L_{rec}. Where I_t^{HR} denotes the tth reconstructed image, and I_t^{GT} denotes the tth real image, ϵ empirically set to 1×10^{-3}.

$$L_{rec} = \sqrt{\left\| I_t^{HR} - I_t^{GT} \right\|^2 + \epsilon^2} \tag{11}$$

4 Experiments

In this section, we first introduce the datasets and implementation details used in the experiments. Next, we compare our proposed network with the SOTA methods. Finally, we conduct ablation studies to evaluate our framework.

4.1 Experimental Setup

Dataset. Due to the lack of auxiliary features in the existing recognized STSR training datasets, it cannot meet the input requirements of our proposed network. We used the Tungsten rendering engine [35] to render image sequences from nine public scenes [36] as the datasets. The entire datasets has 1016 sets of rendered images. Each set contains seven consecutive frames of rendered images and their corresponding auxiliary feature images (albedo, depth, normal). The image resolution is $512 * 512$. Among them, 776 sets of rendered sequences are used for training, 80 sets of rendered sequences are used for validation, and 160 sets of rendered sequences are used for testing.

This paper mainly focuses on the performance of the network to achieve single-frame interpolation and 4x super-resolution. Therefore, the input rendered image sequences remove the three discontinuous frames in the middle (the second frame, the fourth frame, and the sixth frame) and performs 4x downsampling. In summary, the specific input content of our network is shown in Table 1.

Implementation Details. During training, we randomly horizontal-flipping, and rotating 90°, 180°, and 270°, and cropping the input content for data augmentation. And Adam optimizer [37] is used to train our network by setting

Table 2. Quantitative results of our network with other SOTA methods in dining room and kitchen scenes. Best results are shown in boldface.

Method	Dining Room		Kitchen		Speed	Parameters
VFI+(V)SR	PSNR	SSIM	PSNR	SSIM	FPS	(Millions)
SuperSloMo+Bicubic	23.48	0.7837	26.00	0.8458	-	19.8
SuperSloMo+EDVR	24.84	0.8318	26.66	0.8480	16.35	19.8+20.7
SuperSloMo+RBPN	25.03	0.8378	26.78	0.8510	12.07	19.8+12.7
SuperSloMo+BasicVSR	29.63	0.9251	30.50	0.9265	14.43	19.8+6.3
DAIN+Bicubic	23.51	0.7867	26.05	0.8478	-	24.0
DAIN+EDVR	25.01	0.8380	26.78	0.8507	15.60	24.0+20.7
DAIN+RBPN	26.14	0.8927	28.07	0.9020	11.81	24.0+12.7
DAIN+BasicVSR	30.16	0.9408	30.72	0.9320	13.61	24.0+6.3
RIFE+Bicubic	23.50	0.7837	25.99	0.8450	-	9.8
RIFE+EDVR	25.03	0.8378	26.78	0.8510	17.84	9.8+20.7
RIFE+RBPN	26.06	0.8907	27.99	0.9022	12.86	9.8+12.7
RIFE+BasicVSR	30.20	0.9402	30.69	0.9325	22.51	9.8+6.3
STARnet	29.66	0.9070	30.41	0.9011	29.38	111.61
Zooming Slow-Mo	30.75	0.9495	31.04	0.9425	**30.30**	11.10
TMNet	29.85	0.9406	30.36	0.9276	28.87	12.26
RSTT	31.69	0.9605	32.33	0.9534	30.04	7.67
Ours	**33.38**	**0.9662**	**33.83**	**0.9672**	23.12	**2.41**

$\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is decayed from 1×10^{-4} to 1×10^{-7}. Furthermore, our network is implemented with PyTorch and trained on the NVIDIA RTX-3060 GPUs.

Evaluation Metrics. When evaluating the performance of the network, we use Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) [38] as evaluation metrics of image quality. The unit of PSNR is dB. The larger the value, the smaller the difference between the reconstructed images and the real images. The larger the SSIM value, the higher the similarity between the reconstructed images and the real images.

4.2 Comparison to State-of-the-Art Methods

Quantitative comparisons. Since our network achieves both interpolation and super-resolution tasks, we compare our network with existing one-stage and two-stage STVSR methods. We select four SOTA one-stage STVSR networks: STARnet [28], Zooming Slow-Mo [10], TMNet [31] and RSTT [11]. For the two-stage methods, we select three SOTA VFI networks: SuperSloMo [15], DAIN [16], RIFE [8], and three SOTA VSR networks: EDVR [27], RBPN [20], BasicVSR [6]. In addition, we also use the classic Bicubic method to achieve image super-resolution.

We test the performance of all networks using 160 sequences. The 160 test sets contain all the scenes we use to render image sequences. We also compare the networks inference time in Frame Per Second (FPS) and networks size in terms of the number of parameters.

Table 2 shows the quantitative results of our network with other SOTA methods for rendered sequences of dining room and kitchen scenes. The experimental results demonstrates that compared with other SOTA methods, our network has better performance in achieving the joint task of interpolation and super-resolution for rendered sequences. And because our network additionally inputs the HFR and HR auxiliary feature sequences corresponding to the target sequences, it does not need to design complex modules to estimate the motion state of the intermediate frames, which greatly reduces the network parameters. However, this additional input also causes our network to spend more time processing the input, reducing processing speed. Fortunately, although the processing speed of our network is lower than that of the one-stage methods, the processing results are significantly better than other methods (the PSNR value is improved by at least 1.5dB, and the SSIM value is improved by at least 0.0057). At the same time, our network is still efficient compared to the time spent rendering.

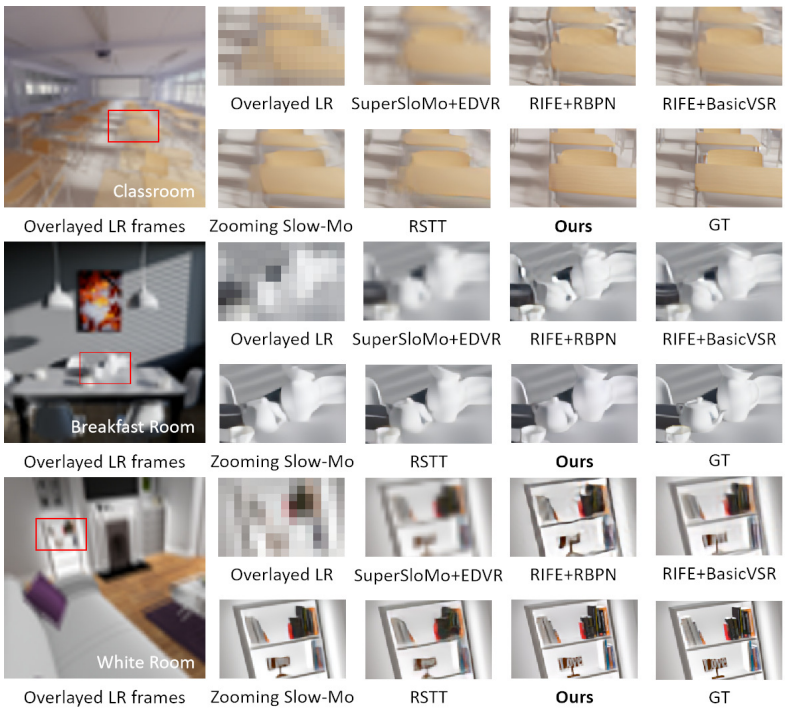

Fig. 5. Qualitative results of our network with other SOTA methods in classroom, breakfast room and white room scenes.

Table 3. Quantitative results of ablation study.

	BNetv1	BNetv2	BNetv3	BNetv4	BNetv5
HFR Auxiliary Features		√	√	√	√
Weighted Aggregation			√	√	√
LR Auxiliary Features				√	
HR Auxiliary Features					√
PSNR	29.40	30.83	31.37	32.67	33.83
SSIM	0.9105	0.9379	0.9399	0.9571	0.9672

Qualitative comparisons. The qualitative comparisons of our network with other SOTA STVSR networks are shown in Fig. 5. To show the effect of the networks on different problems, we chose three different scenes (Classroom, Breakfast Room and White Room).

The classroom scene shows the ability of the networks to infer the motion state of the intermediate frames. It can be found that the motion state of the intermediate frames generated by our network is closer to the real motion state, and the motion edge part is clearer than other methods. The breakfast room scene shows the processing power of the networks when facing multiple objects with similar colors and adjacent objects. Although there are some gaps between the results generated by our network and the real state, compared with other methods, our network can obtain clearer boundaries in the contact part of similar objects. The white room scene shows the ability of the networks to recover objects textures, details when achieving super-resolution task. It can be seen intuitively that the textures and details restored by our network are clearer than those of other methods and are closer to the real textures and details.

4.3 Ablation Study

To verify the effect of the proposed modules in our network, the comprehensive ablation studies are conducted in this section.

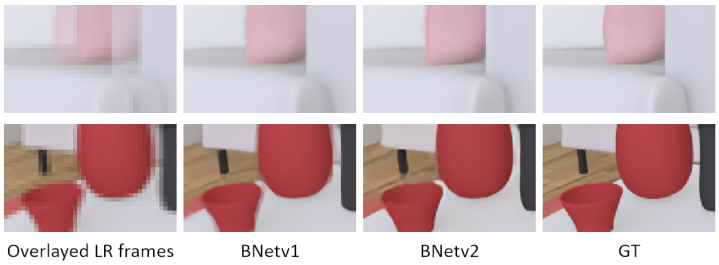

Fig. 6. Ablation study on AFGI.

Effectiveness of AFGI. To study the AFGI module, we design two baseline modules, BNetv1 and BNetv2. Unlike BNetv2, which uses the AFGI module to synthesize intermediate frames, BNetv1 removes all parts about auxiliary features and only uses deformable alignment to warp the input frames to the intermediate moments to obtain the intermediate frames. Moreover, BNetv1 and BNetv2 directly fuse the incoming information when aggregating contextual information and do not add additional auxiliary features. It can be seen from Table 3 that the PSNR of BNetv2 is 1.43dB higher than that of BNetv1, and the SSIM is improved by 0.0274. From the qualitative results shown in Fig. 6, we can see that the results generated by BNetv1 are relatively fuzzy on the surface and boundary of moving objects, while BNetv2 introduces corresponding auxiliary features, which provides the network with the real motion state of the intermediate frames, making the results it generates clearer. The comparison results show that introducing the HFR auxiliary feature sequences corresponding to the target sequences into the network can effectively help the network generate more accurate intermediate frames.

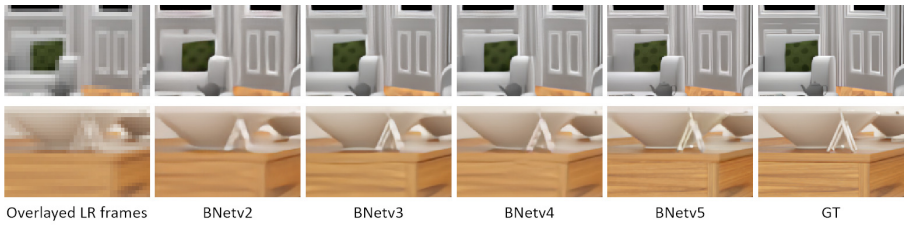

Fig. 7. Ablation study on WAUpS.

Effectiveness of WAUpS. To verify the effect of the proposed WAUpS module, we compare the aggregation contextual information part based on BNetv2. BNetv5 directly uses the WAUpS module. BNetv3 removes the auxiliary features and only uses weighted aggregation. BNetv4 changes the HR auxiliary features used in BNetv5 to LR. They all use bidirectional propagation to aggregate contextual information. As can be seen from Table 3, the PSNR of BNetv3 is 0.54dB higher than that of BNetv2, and the SSIM of BNetv3 is 0.002B higher than that of BNetv2. The PSNR of BNetv4 is 1.3dB higher than that of BNetv3, and the SSIM of BNetv4 is 0.0172B higher than that of BNetv3. The PSNR of BNetv5 is 1.16dB higher than that of BNetv4, and the SSIM of BNetv5 is 0.0101B higher than that of BNetv4. Fig. 7 shows the qualitative results of the four variant networks. It can be found that the images generated by BNetv5 are clearer in details and textures than those generated by the other three networks. Among them, the images generated by BNetv3 have clearer details than those generated by BNetv2, and the images generated by BNetv4 have clearer details than those generated by BNetv3. From the comparison results, we can see that using

similarity to selectively aggregate information can effectively avoid the influence of irrelevant information on the results. In addition, the introduction of auxiliary features can also provide the network with richer detail information, and the closer the resolution of the auxiliary features is to the target resolution, the more helpful the information provided is for the processing results.

5 Conclusion

In this paper, we propose an end-to-end spatial-temporal super-resolution network for rendered sequences. Among them, the AFGI module uses the auxiliary features corresponding to the target frames to guide the synthesis of more accurate intermediate frames. The WAUpS module selectively integrates contextual information according to the correlations between the incoming information and the current frames, and integrates the corresponding HR auxiliary features after upsampling to add more details. Experimental results show that by introducing the HFR and HR auxiliary feature sequences corresponding to the target sequences, and through the processing of AFGI and WAUpS modules, our network can obtain high-quality rendered sequences post-processing results and can help improve rendering efficiency.

Acknowledgments. The numerical calculations in this paper have been done on the supercomputing system in the High Performance Computing Center of Southwest Petroleum University.

References

1. Kajiya, J.T.: The rendering equation. In: Proceedings of conference on Computer graphics and interactive techniques. pp. 143–150 (1986)
2. Bako, S., Vogels, T.: Kernel-predicting convolutional networks for denoising monte carlo renderings. ACM Transactions on Graphics (TOG) **36**(4), 97–1 (2017)
3. Hou, Q., Li, Z., Marshall, C.S., et al.: Fast monte carlo rendering via multi-resolution sampling. arXiv preprint arXiv:2106.12802 (2021)
4. Wei, X.Y., Huang, H.Z., Shi, Y.J., et al.: End-to-end adaptive monte carlo denoising and super-resolution. arXiv preprint arXiv:2108.06915 (2021)
5. Lu, Y.F., Fu, S.Y., Zhang, X.H., et al.: Denoising monte carlo renderings via a multi-scale featured dual-residual gan. The Visual Computer **37**(9-11), 2513–2525 (2021)
6. 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)
7. Liu, C., Yang, H., Fu, J., Qian, X.: Learning trajectory-aware transformer for video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5687–5696 (2022)
8. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: European Conference on Computer Vision. pp. 624–642. Springer (2022)

9. Plack, M., Briedis, K.M., Djelouah, A., Hullin, M.B., Gross, M., Schroers, C.: Frame interpolation transformer and uncertainty guidance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9811–9821 (2023)
10. Xiang, X., Tian, Y., Zhang, Y., Fu, Y., Allebach, J.P., Xu, C.: Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3370–3379 (2020)
11. Geng, Z., Liang, L., Ding, T., Zharkov, I.: Rstt: Real-time spatial temporal transformer for space-time video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17441–17451 (2022)
12. Back, J., Hua, B.S., Hachisuka, T., Moon, B.: Self-supervised post-correction for monte carlo denoising. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–8 (2022)
13. Alzbier, A.M.T., Chen, C., Shen, Z., Zhang, R.: Fast dual deep convolutional autoencoder network for denoising monte carlo rendering. Journal of Imaging Science & Technology **67**(3) (2023)
14. He, R., Zhou, S., Sun, Y., Cheng, R., Tan, W., Yan, B.: Low-latency space-time supersampling for real-time rendering. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 2103–2111 (2024)
15. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9000–9008 (2018)
16. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3703–3712 (2019)
17. Li, H., Yuan, Y., Wang, Q.: Video frame interpolation via residue refinement. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2613–2617. IEEE (2020)
18. Lee, H., Kim, T., Chung, T.y., Pak, D., Ban, Y., Lee, S.: Adacof: Adaptive collaboration of flows for video frame interpolation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5316–5325 (2020)
19. Cheng, X., Chen, Z.: Video frame interpolation via deformable separable convolution. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10607–10614 (2020)
20. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3897–3906 (2019)
21. Irani, M., Peleg, S.: Improving resolution by image registration. CVGIP: Graphical models and image processing **53**(3), 231–239 (1991)
22. Irani, M., Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency. Journal of visual communication and image representation **4**(4), 324–335 (1993)
23. Chan, K.C., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5972–5981 (2022)
24. 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)

25. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9308–9316 (2019)
26. 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)
27. 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)
28. Haris, M., Shakhnarovich, G., Ukita, N.: Space-time-aware multi-resolution video enhancement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2859–2868 (2020)
29. Wang, H., Xiang, X., Tian, Y., Yang, W., Liao, Q.: Stdan: deformable attention network for space-time video super-resolution. IEEE Transactions on Neural Networks and Learning Systems (2023)
30. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems **28** (2015)
31. Xu, G., Xu, J., Li, Z., Wang, L., Sun, X., Cheng, M.M.: Temporal modulation network for controllable space-time video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6388–6397 (2021)
32. Chen, K., Yue, Z., Shi, M.: Space-time video super-resolution using long-term temporal feature aggregation. Autonomous Intelligent Systems **3**(1), 5 (2023)
33. 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)
34. 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)
35. Bitterli, B., Rousselle, F., Moon, B., Iglesias-Guitián, J.A., Adler, D., Mitchell, K., Jarosz, W., Novák, J.: Nonlinearly weighted first-order regression for denoising monte carlo renderings. In: Computer Graphics Forum. vol. 35, pp. 107–117. Wiley Online Library (2016)
36. Bitterli, B.: Rendering resources (2016), https://benedikt-bitterli.me/resources/
37. Diederik, P.K.: Adam: A method for stochastic optimization. (No Title) (2014)
38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)