Space-Time Video Super-Resolution Using Temporal Profiles

Zeyu Xiao, Zhiwei Xiong, Xueyang Fu, Dong Liu, Zheng-Jun Zha
University of Science and Technology of China
zwxiong@ustc.edu.cn

ABSTRACT

In this paper, we propose a novel space-time video super-resolution method, which aims to recover a high-frame-rate and high-resolution video from its low-frame-rate and low-resolution observation. Existing solutions seldom consider the spatial-temporal correlation and the long-term temporal context simultaneously and thus are limited in the restoration performance. Inspired by the epipolar-plane image used in multi-view computer vision tasks, we first propose the concept of *temporal-profile* super-resolution to directly exploit the spatial-temporal correlation in the long-term temporal context. Then, we specifically design a feature shuffling module for spatial retargeting and spatial-temporal information fusion, which is followed by a refining module for artifacts alleviation and detail enhancement. Different from existing solutions, our method does not require any explicit or implicit motion estimation, making it lightweight and flexible to handle any number of input frames. Comprehensive experimental results demonstrate that our method not only generates superior space-time video super-resolution results but also retains competitive implementation efficiency.

CCS CONCEPTS

• **Computing methodologies → Computational photography**.

KEYWORDS

temporal profile, video frame interpolation, video super-resolution

ACM Reference Format:
Zeyu Xiao, Zhiwei Xiong, Xueyang Fu, Dong Liu, Zheng-Jun Zha . 2020. Space-Time Video Super-Resolution Using Temporal Profiles. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413667

1 INTRODUCTION

With the increasing usage of mobile phones and digital cameras, acquiring videos becomes cheaper and easier. High-frame-rate (HFR) and high-resolution (HR) videos are desired in various applications, such as film making and high-definition television. Therefore, converting low-frame-rate (LFR) and low-resolution (LR) videos to HFR and HR versions is critical for enhancing the visual quality of captured videos.

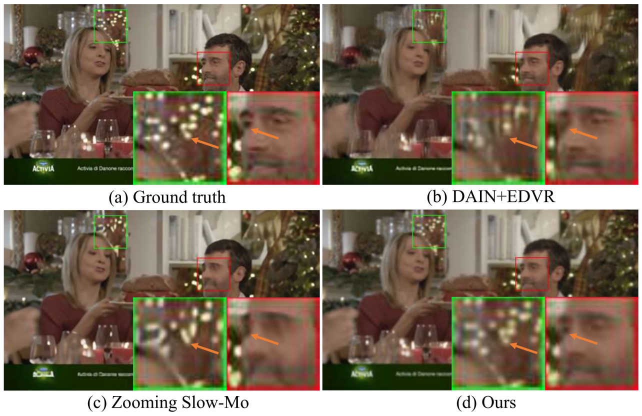

(a) Ground truth (b) DAIN+EDVR
(c) Zooming Slow-Mo (d) Ours

Figure 1: Examples of space-time video super-resolution. We show the ground truth, cascading of two representative video frame interpolation and video super-resolution methods (DAIN [1] + EDVR [36]), one-stage model (Zooming Slow-Mo [39]) and our results. Orange arrows indicate where our model generates vivid details and less artifacts.

To achieve the above goal, a straightforward strategy is to cascade video frame interpolation (VFI) and video super-resolution (VSR) techniques. The VFI methods [21, 24] aim to recover unseen latent intermediate frames from captured ones, which can up-convert frame rate and improve visual quality, while the VSR methods [16] can utilize temporal information of the consecutive frames to enhance the spatial resolution. However, directly cascading VFI and VSR is sub-optimal since it cannot fully exploit the spatial-temporal correlation in videos. Moreover, while the computational efficiency is low, it is easy to introduce cumulative errors. As shown in Figure 1, cascading of two representative VFI and VSR methods tends to synthesize blurring results.

Space-time video super-resolution (STVSR) has been studied as a challenging inverse problem [8, 18, 23, 30, 31] before the deep learning era, although the reconstruction performance is limited due to the lack of priors learned from large data. Recently, Xiang *et al.* [39] propose a one-stage deep learning framework, named Zooming Slow-Mo, to address VFI and VSR simultaneously. Their method can be regarded as a pioneer work along this line and achieves promising performance, yet it still has the following deficiencies. First, using deformable ConvLSTM to implicitly align frames may miss the long-term temporal context, since more complex frame alignment rules need to be designed when more frames are involved. Second, Zooming Slow-Mo sometimes generates unrealistic artifacts as shown in Figure 1.

Inspired by the epipolar-plane image (EPI) [2] widely used in multi-view computer vision tasks, we propose a STVSR method

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '20, October 12–16, 2020, Seattle, WA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413667

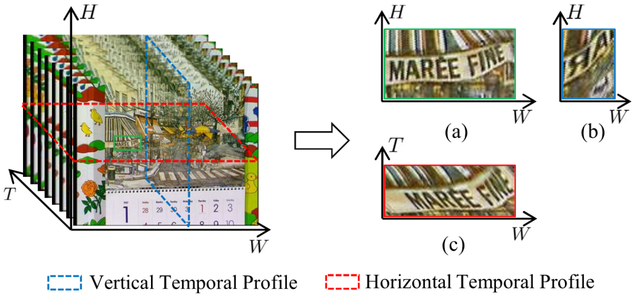


Figure 2: Example of how video frames are converted to TPs. We show (a) a patch of the frame, (b) vertical TP of the patch and (c) horizontal TP of the patch. Both horizontal and vertical TPs maintain similar structures to those in the spatial domain.

based on the *temporal profile* (TP) to address VFI and VSR simultaneously. In Figure 2, we give an example to illustrate how video frames are converted to TPs and how TPs look like. It can be seen that TPs maintain similar structures to those in the spatial domain, which makes it possible to perform super-resolution directly on TPs. There are several benefits for doing so: (1) STVSR can be effectively modeled as a learning-based restoration task focusing on the specific 2D structure of TPs; (2) since TPs contain both space and time dimensions, spatial-temporal correlation can be better exploited; and (3) compared with existing methods relying on multi-frame alignment, long-term temporal context can be integrated by TPs in a more flexible way.

Based on the above observation, we explicitly use TPs for STVSR in this paper. The examples shown in Figure 1 demonstrate that our method achieves improved visual quality in comparison with existing solutions.

In summary, the contributions of this work are three-fold:

- We introduce a new perspective for STVSR by exploiting the spatial-temporal correlation in the form of TPs. To the best of our knowledge, this is the first effort to solve STVSR in a transformed domain inspired by multi-view computer vision tasks.
- We propose a TP-based STVSR network to simultaneously address VFI and VSR, which consists of three elaborately designed modules. The proposed network has the advantages of end-to-end training, high computational efficiency, and lightweight architecture.
- Compared with existing solutions including the state-of-the-art, our method generates superior results both quantitatively and qualitatively. It also has better generalization ability and can be readily applied to real-world scenarios such as old movie restoration.

2 RELATED WORK

2.1 Epipolar-Plane Image and Temporal Profile

The epipolar-plane image (EPI), a well-known term in multi-view computer vision tasks, describes a static scene from a dense sequence of images in a cross-dimensional way [2]. EPI has proven its effectiveness in light field scene geometry inferring [27] and light field super-resolution [4, 5, 38]. Given a 4D light field $L(x, y, u, v)$, where (x, y) denote the spatial dimensions and (u, v) denote the angular dimensions, one can produce a slice $E_{y^*,v^*}(x, u)$ (or $E_{x^*,u^*}(y, v)$), by gathering the light field samples with fixed spatial coordinate y^* and angular coordinate v^* (or x^* and u^*). This resulting slice is the so-called EPI, which represents the 4D light field in a different way from the common view-wise images denoted as $I_{u^*,v^*}(x, y)$.

Considering a video denoted as a 3D volume $V(w, h, t)$, where w, h and t denote width, height, and time dimensions, the definition of TP is similar to EPI. For instance, the vertical TP $P_{w^*}(h, t)$ and the horizontal TP $P_{h^*}(w, t)$ are the slices generated when $w = w^*$ and $h = h^*$. Although TP has been widely used as a visual metirc for the evaluation of video reconstruction results in VFI and VSR [3, 42], it has not been explicitly applied to the video reconstruction process. In this paper, by utilizing TPs to exploit the spatial-temporal correlation in the long-term temporal context, we model STVSR from a new perspective.

2.2 Video Frame Interpolation (VFI) and Video Super-Resolution (VSR)

VFI aims to recover unseen intermediate frames to up-convert frame rate and improve visual quality of the captured videos. Existing deep-learning-based VFI methods can be categorized into flow-based and kernel-based ones. With the advances in optical flow estimation [7, 12, 28], several approaches either predict bidirectional flow [40] or use the bilinear warping operation to align input frames based on linear motion models [13, 21, 35]. On the other hand, VFI can be formulated as convolutional operations over local patches [24, 25]. As a representative work along this line, Niklaus *et al.* [24] propose the AdaConv to estimate spatially-adaptive convolutional kernels for each output pixel.

VSR emerges as an adaptation of single-image super-resolution techniques by exploiting additional information from neighboring frames. Early VSR methods adopt an explicit motion compensation module to align different frames, among which Tao *et al.* [34] introduce a sub-pixel motion compensation layer to jointly perform motion compensation and up-sampling. On the other hand, implicit motion compensation schemes have also been widely adopted. For instance, Jo *et al.* [14] propose a network by using dynamic up-sampling filters, while Wang *et al.* [36] achieve deformable alignment through the pyramid, cascading and deformable structure.

An intuitive solution to address STVSR is to directly cascade VFI and VSR in a two-stage manner. However, this cascading approach may accumulate errors and cause unexpected artifacts. In contrast, as an end-to-end framework, our proposed method based on TPs is more effective and efficient, which can better recover temporal and spatial details for STVSR.

2.3 Space-Time Video Super-Resolution

To increase resolution in both space and time dimensions of videos, Shechtman *et al.* [30] design a space-time smoothness regularization method as a pioneer work for STVSR. Later, Mudenagudi *et al.* [23] pose STVSR as a reconstruction problem and use the maximum a posteriori-Markov Random Field with graph-cuts to solve it. Recently, Xiang *et al.* [39] propose a one-stage deep-learning-based

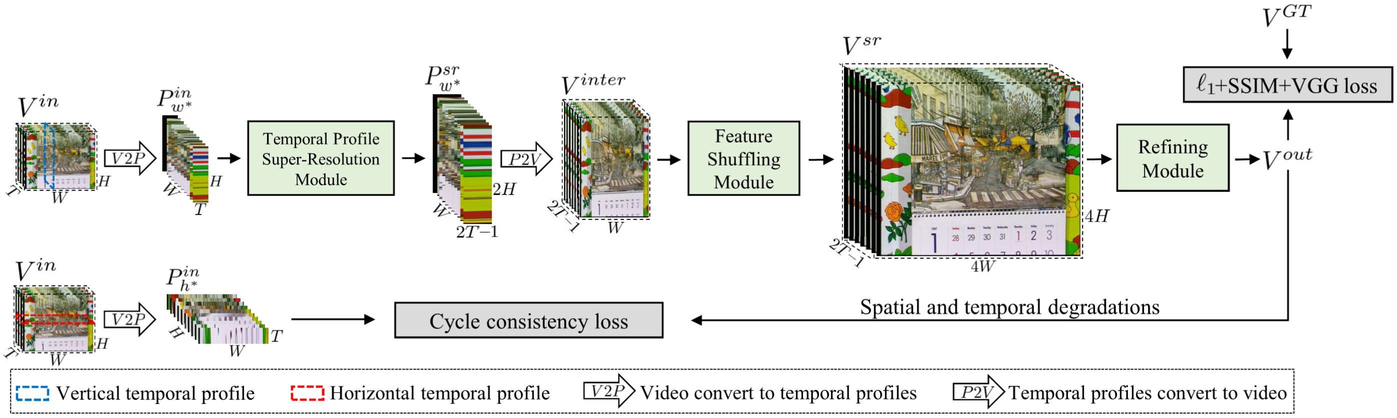

V^{in}
V2P
$P^{in}_{w^*}$
Temporal Profile Super-Resolution Module
$P^{sr}_{w^*}$
P2V
V^{inter}
Feature Shuffling Module
V^{sr}
Refining Module
V^{out}
V^{GT}
ℓ_1+SSIM+VGG loss
H
W
T
2H
2T−1
4H
4W
$P^{in}_{h^*}$
Cycle consistency loss
Spatial and temporal degradations
Vertical temporal profile
Horizontal temporal profile
V2P Video convert to temporal profiles
P2V Temporal profiles convert to video

Figure 3: Overview of our proposed network for STVSR, where we take ×2 VFI and ×4 VSR as an example.

framework which consists of three main modules: frame feature temporal interpolation, deformable ConvLSTM and frame reconstruction. However, the limited temporal context and unrealistic artifacts are still drawbacks of this method.

Different from [39], we employ TPs to capture the spatial-temporal correlation without the need of motion compensation. This makes our method lightweight and flexible for any number of input frames. In addition, we also utilize a variety of complementary loss terms in the network training, which further promotes the reconstruction performance.

3 PROPOSED METHOD

Given a LFR and LR video clip $V^{in} \in \mathbb{R}^{W\times H\times T}$, our goal is to generate a HFR and HR video clip $V^{out} \in \mathbb{R}^{aW\times aH\times(bT-b+1)}$, which can provide a clearer and smoother visual experience. Here W, H and T refer to width, height, and number of frames, while a and b denote the magnification factors in space and time dimensions. Without loss of generality, we assume $a = 4$ and $b = 2$ throughout this paper, and other magnification factors can be easily realized under the same framework. Figure 3 shows the overall structure of our proposed network for STVSR. As can be seen, our network consists of three parts: Temporal-Profile Super-Resolution Module (TPSRM), Feature Shuffling Module (FSM) and Refining Module (RM). In general, we first convert the LFR and LR video clip into TPs and send them to TPSRM to generate the super-resolved TPs with the target frame rate. Then, we convert the super-resolved TPs back to the video domain and send them to FSM to generate the video clip with the target spatial resolution. Finally, to obtain temporally-smooth and spatially-clear sequences, we further utilize RM for artifacts alleviation and detail enhancement.

3.1 Temporal-Profile Super-Resolution Module

To effectively capture the spatial-temporal correlation contained in TPs, we use the advanced Information Multi-distillation Network (IMDN) [11] as the backbone of TPSRM, which super-resolves TPs with low computational cost and memory usage[1]. After converting the input video into W vertical TPs denoted as $P^{in}_{w^*} \in \mathbb{R}^{H\times T}$, the super-resolved vertical TPs $P^{sr}_{w^*} \in \mathbb{R}^{2H\times(2T-1)}$ are generated through TPSRM as[2]:

$$P^{sr}_{w^*} = N_{\text{TPSRM}}\left(P^{in}_{w^*}\right), \quad w^* = 1, \ldots, W, \tag{1}$$

where $N_{\text{TPSRM}}(\cdot)$ denotes the processing of TPSRM. This module is optimized by the ℓ_1 loss. Specifically, given a training set $\left\{P^{in}_{w^*}, P^{de}_{w^*}\right\}_{w^*=1}^{W_{train}}$, the loss function for training TPSRM is

$$\mathcal{L}_{\text{TPSRM}}\left(\Theta_{tpsrm}\right) = \frac{1}{W_{train}} \sum_{w^*=1}^{W_{train}} \left\| N_{\text{TPSRM}}\left(P^{in}_{w^*}\right) - P^{de}_{w^*} \right\|_1, \tag{2}$$

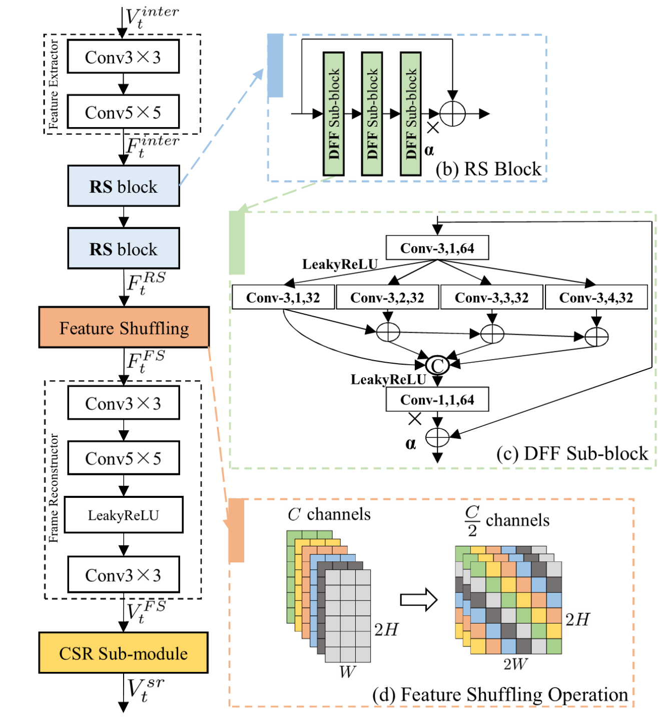
where Θ_{tpsrm} denotes the learnable parameter set, W_{train} is the total number of extracted TPs from training video clips, $P^{de}_{w^*} \in \mathbb{R}^{2H\times(2T-1)}$ denote the vertical TPs converted from HFR and HR training videos with corresponding spatial degradation (*i.e.*,1/2 in height and 1/4 in width), and W_{train} is the total number of TPs used for training. Note that, while we use vertical TPs here, converting videos into horizontal TPs as input of TPSRM makes no difference in principle.

3.2 Feature Shuffling Module

After converting the super-resolved vertical TPs back to the video domain, we obtain an intermediate result $V^{inter} \in \mathbb{R}^{W\times 2H\times(2T-1)}$ that has the target frame rate but not the target spatial resolution. To complete the HFR and HR video, we specifically design a feature shuffling module, which consists of one feature extractor, two residual stacking (RS) blocks, one feature shuffling operator, one frame reconstructor and one cascaded super-resolution (CSR) sub-module. Since V^{inter} and V^{out} have the same size in the time dimension, the FSM turns into a retargeting process in the space dimensions

[1] Note that, the focus of this work is not designing a single-image super-resolution network, and IMDN can be readily replaced by other advanced embodiments.

[2] To comply with the general configuration in VFI, we cut the temporal resolution from 2T to 2T-1 after super-resolution of TPs.

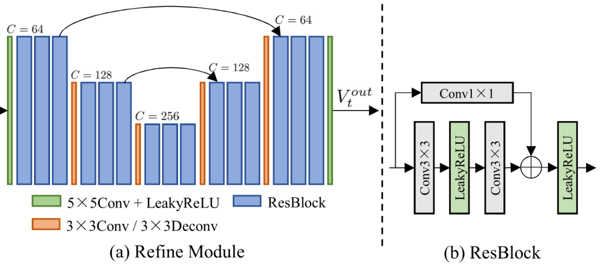


(a) Feature Shuffling Module

Figure 4: The detailed structure of Feature Shuffling Module. © and ⊕ denote concatenation and element-wise addition, respectively. The three parameters after "Conv" in (c) represent convolutional kernel size, dilation rate and the number of channels, respectively. α is the residual scaling parameter.

(*i.e.*, ×2 in height and ×4 in width). As shown in Figure 4(a), we first use the feature extractor with two convolutional layers to extract feature maps for each individual frame $V_t^{inter} (t = 1, \dots, 2T-1)$, denoted as $F_t^{inter} \in \mathbb{R}^{W \times 2H \times C}$, where C is the number of channels (an even number). To better fuse spatial-temporal information, we design the RS block to generate hierarchical feature representations, which are then fed into the feature shuffling operator followed by the frame reconstructor and the CSR sub-module to obtain the target HR and HFR video frames. Below we describe each component in detail.

Residual Stacking Block. Since video frames contain spatial information at different scales, using multi-scale convolutions to extract features is of great benefit for video reconstruction tasks. However, directly increasing the size of convolution kernels will increase the amount of parameters and thus storage resources. We instead adopt dilated convolutions [41] to extract multi-scale features. By dilating the same filter to different scales, dilated convolutions can increase the contextual area without introducing extra parameters. Furthermore, we reinforce the representation ability by introducing a hierarchical addition scheme realized by the dilated feature fusion (DFF) sub-block. As shown in Figure 4(c), after extracting features with a single convolutional layer, we hierarchically add the feature maps obtained using kernels of different dilation rates before concatenating them. Here we adopt four different dilation rates. After collecting multi-scale features, we fuse them through a 1×1 convolution layer followed by the LeakyReLU activation. In our implementation, three stacked DFF sub-blocks and skip connections with residual scaling [19, 37] make up one RS block, as shown in Figure 4(b). The output of two RS blocks is denoted as $F_t^{RS} \in \mathbb{R}^{W \times 2H \times C}$.



(a) Refine Module

(b) ResBlock

Figure 5: The detailed structure of Refining Module. ⊕ means element-wise addition and C denotes the number of channels.

Feature Shuffling Operator. The goal of feature shuffling is to double the size of the feature map in the width dimension and keep the height dimension unchanged. Inspired by pixel-shuffle [32], we design a feature shuffling operator which generates feature maps $F_t^{FS} \in \mathbb{R}^{2W \times 2H \times \frac{C}{2}}$ from $F_t^{RS} \in \mathbb{R}^{W \times 2H \times C}$. This periodic shuffling operator in the feature domain rearranges the elements of a feature tensor as illustrated in Figure 4(d).

Cascaded Super-Resolution Sub-module. The CSR sub-module aims to perform ×2 super-resolution on the space dimensions. We use a frame reconstructor which consists of several convolution layers and LeakyReLU before CSR sub-module to reconstruct individual video frames $V_t^{FS} \in \mathbb{R}^{2W \times 2H}$ from the feature maps $F_t^{FS} \in \mathbb{R}^{2W \times 2H \times \frac{C}{2}}$. To generate the video frames $V_t^{sr} \in \mathbb{R}^{4W \times 4H}$ with the target spatial resolution, we again adopt IMDN [11] as the backbone of CSR.

In summary, the processing of FSM can be expressed as

$$V_t^{sr} = N_{\mathrm{FSM}} \left(V_t^{inter} \right), t = 1, \dots, 2T-1. \tag{3}$$

This module is optimized by the ℓ_1 loss. Specifically, given a training set $\left\{ V_t^{inter}, V_t^{GT} \right\}_{t=1}^{T_{train}}$, the loss function for training FSM is

$$\mathcal{L}_{\mathrm{FSM}} \left(\Theta_{fsm} \right) = \frac{1}{T_{train}} \sum_{t=1}^{T_{train}} \left\| N_{\mathrm{FSM}} \left(V_t^{inter} \right) - V_t^{GT} \right\|_1, \tag{4}$$

where Θ_{fsm} is the learnable parameter set, $V_t^{GT} \in \mathbb{R}^{4W \times 4H}$ denote the ground truth HFR and HR training video frames, and T_{train} is the total number of frames used for training.

3.3 Refining Module

To further alleviate possible artifacts introduced by the former two modules and enhance spatial-temporal details, we design a simple yet effective RM based on the U-Net structure [29], as shown in

Figure 5. RM consists of one contracting path and one expansive path with skip connection. Before sending video frames into RM, we use a convolution layer with 5 × 5 kernel size to extract features for refinement. Both paths adopt stacked ResBlocks [9] as backbones. We use 15 ResBlocks in RM and a convolution layer with 5 × 5 kernel size for the final reconstruction. The processing of RM can be expressed as

$$V_t^{out} = N_{\text{RM}}\left(V_t^{sr}\right), \quad t = 1, \ldots, 2T-1, \tag{5}$$

where $V_t^{out} \in \mathbb{R}^{4H \times 4W}$ is the refined HR and HFR video frames. To optimize RM, we combine ℓ_1 loss, SSIM loss, VGG loss and cycle consistency loss.

ℓ_1 **loss** is defined as

$$\mathcal{L}_{\text{RM}}^{\ell_1} = \frac{1}{T_{train}} \sum_{t=1}^{T_{train}} \left\| N_{\text{RM}}\left(V_t^{sr}\right) - V_t^{GT} \right\|_1 . \tag{6}$$

SSIM loss [10] is defined as

$$\mathcal{L}_{\text{RM}}^{SSIM} = \frac{1}{T_{train}} \sum_{t=1}^{T_{train}} \left(1 - \text{SSIM}\left(N_{\text{RM}}\left(V_t^{sr}\right), V_t^{GT}\right)\right). \tag{7}$$

VGG loss encourages similar feature representations between the restored frame and the target one [15]. It is calculated on multiple layers of a pre-trained VGG-19 network as

$$\mathcal{L}_{\text{RM}}^{VGG} = \frac{1}{T_{train}} \sum_{\text{t=1}}^{T_{train}} \sum_{j=1,3,5} \left\| \phi_j\left(N_{\text{RM}}\left(V_t^{sr}\right)\right) - \phi_j\left(V_t^{GT}\right) \right\|_2^2 , \tag{8}$$

where ϕ_j denotes the feature map at the j-th layer of the VGG-19 network.

Cycle consistency loss is adopted to ensure the spatial-temporal consistency between the reconstructed video and its LFR and LR input. At the same time, using cycle consistency loss can avoid over-enhancement. Since TPSRM is designed for super-resolving vertical TPs, here we use horizontal TPs for cycle consistency loss calculation. Specifically, we measure the difference of horizontal TPs converted from the degraded output and the given input as

$$\mathcal{L}_{\text{RM}}^{Cycle} = \frac{1}{H_{train}} \sum_{h^*=1}^{H_{train}} \left\| P_{h^*}^{de} - P_{h^*}^{in} \right\|_1 , \tag{9}$$

where $P_{h^*}^{de} \in \mathbb{R}^{W \times T}$ denote the horizontal TPs converted from the reconstructed video $\text{V}^{out} \in \mathbb{R}^{4W \times 4H \times (2T-1)}$ after spatial and temporal degradations (*i.e.*, 1/4 in height, 1/4 in width, and 1/2 in frame rate), $P_{h^*}^{in} \in \mathbb{R}^{W \times T}$ denote the horizontal TPs converted from the input video $V^{in} \in \mathbb{R}^{W \times H \times T}$, and H_{train} is the total number of TPs used for training.

Therefore, the total loss function for RM is

$$\mathcal{L}_{\text{RM}}\left(\Theta_{rm}\right) = \mathcal{L}_{\text{RM}}^{\ell_1} + \lambda_1 \mathcal{L}_{\text{RM}}^{SSIM} + \lambda_2 \mathcal{L}_{\text{RM}}^{VGG} + \lambda_3 \mathcal{L}_{\text{RM}}^{Cycle}, \tag{10}$$

where Θ_{rm} denotes the learnable parameter set, and λ_1, λ_2, and λ_3 are weighting factors.

4 EXPERIMENTS AND ANALYSIS

4.1 Implementation Details

Training Set. We use the same experimental setting as [39] and train our model on the Vimeo90K [40] dataset. A total of around 60,000 video clips with spatial resolution of 448 × 256 and frame number of 7 are used for training the network. We take the 4-times bicubic-down-sampled odd-indexed frames as LFR and LR inputs, and the corresponding consecutive HFR and HR sequences as supervision. Rotation and flipping are applied for data augmentation.

Test Setting. We evaluate our method on several testsets. The first one is the Vimeo90K dataset [40] (excluding the training data), which is divided into three subsets: slow motion, medium motion and fast motion. Each subset contains 1225, 4977 and 1613 video clips, respectively. We also use the UCF101 [33] and Vid4 [20] datasets for evaluation of model generalizability. To quantitatively evaluate the reconstructed videos, we choose Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [10], and Naturalness Image Quality Evaluator (NIQE) [22] as metrics.

Training Strategy. Since our network consists of three modules, we adopt a step-by-step training strategy to accelerate training. Specifically, we first train TPSRM according to Equation (2). Then, we freeze its parameters and train FSM based on Equation (4). Finally, we optimize all the three modules together for extra epochs based on Equation (10). In our implementation, we set $\alpha = 0.2$, $\lambda_1 = 0.1$, $\lambda_2 = 0.01$ and $\lambda_3 = 1$.

We utilize the Adam optimizer [17] with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is set to 1, and the initial learning rate is 1e-4. Each module is trained for 4 epochs, and we reduce the learning rate by a factor of 0.2 for every 2 epochs. All experiments are conducted using PyTorch [26] on an NVIDIA GTX1080Ti GPU.

4.2 Comparison to State-of-the-art

We evaluate our method against representative STVSR methods. For the two-stage solutions, we cascade state-of-the-art VFI and VSR methods in different combinations and orders. SepConv [25] and DAIN [1] are selected to perform VFI, while IMDN [11], SAN [6], and EDVR [36] are selected for VSR. Especially, we compare with Zooming Slow-Mo [39], the recently proposed one-stage STVSR method.

Table 1 shows quantitative comparisons on the testsets. As can be seen, our method outperforms the two-stage solutions by a large margin in terms of all the three metrics. In comparison with the one-stage method Zooming Slow-Mo [39], our method achieves overall superior results except on Vimeo90K-Fast. While our PSNR and SSIM values are slightly lower on this testset, we still achieve a notably better NIQE result, which indicates that our method achieves better visual quality.

Exemplar visual results of different methods are shown in Figure 6, where our method achieves notable visual improvements over its competitors. Affected by the cumulative errors, the results generated by the two-stage solutions are generally of poor quality with motion blur. The results from Zooming Slow-Mo are better than the two-stage solutions, however, it tends to produce over-smooth results sometimes. In contrast, our method generates visually appealing video frames with more accurate details and less blurs. As

Table 1: Quantitative comparison of different methods. The best and second results are highlighted in red and blue.

Method		Vimeo90-Slow [40]			Vimeo90K-Medium [40]			Vimeo90K-Fast [40]			Vid4 [20]			UCF101 [33]		
VFI (×2)	VSR (×4)	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓
SepConv [25]	IMDN [11]	31.75	0.8851	7.6781	33.13	0.8986	7.7814	34.31	0.9177	8.5542	24.87	0.7150	6.3421	29.10	0.8790	7.6561
SepConv [25]	SAN [6]	32.12	0.8966	7.1001	33.59	0.9125	7.4623	34.97	0.9194	8.4790	24.93	0.7240	5.8864	29.80	0.8896	7.3087
SepConv [25]	EDVR [36]	32.97	0.9110	7.0023	34.25	0.9240	7.4016	35.51	0.9253	8.4753	25.93	0.7792	5.7024	30.19	0.8994	7.3915
DAIN [1]	IMDN [11]	31.84	0.8878	7.1319	33.39	0.9073	7.5839	34.74	0.9182	8.4278	24.93	0.7197	6.1853	29.57	0.8882	7.2996
DAIN [1]	SAN [6]	32.26	0.8993	7.0546	33.82	0.9249	7.4468	35.27	0.9244	8.4775	25.14	0.7301	5.7853	30.13	0.8990	7.3214
DAIN [1]	EDVR [36]	33.21	0.9126	7.0638	34.73	0.9283	7.3923	35.71	0.9307	8.4696	26.12	0.7856	5.6243	30.54	0.9001	7.4961
VSR (×4)	VFI (×2)	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓
IMDN [11]	SepConv [25]	32.01	0.8867	7.6661	33.22	0.9016	7.6521	34.50	0.9181	8.5417	24.88	0.7155	6.3336	29.12	0.8801	7.4421
IMDN [11]	DAIN [1]	32.27	0.8916	6.9916	33.73	0.9167	7.1657	35.15	0.9206	8.4121	24.99	0.7227	6.2116	29.79	0.8901	7.2246
SAN [6]	SepConv [25]	32.32	0.9006	6.9912	33.73	0.9154	7.3151	35.33	0.9233	8.4221	25.01	0.7313	5.8714	29.92	0.8898	7.3151
SAN [6]	DAIN [1]	32.56	0.9113	6.8954	34.12	0.9284	7.4315	35.47	0.9246	8.3876	25.26	0.7515	6.1654	30.31	0.8992	7.2157
End-to-end Framework		PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓	PSNR↑	SSIM↑	NIQE↓
Zooming Slow-Mo [39]		33.29	0.9127	6.9397	35.24	0.9347	7.3461	36.43	0.9337	8.4093	26.30	0.7975	5.6203	30.90	0.9095	7.2914
Ours		33.40	0.9217	6.1725	35.55	0.9358	6.3704	36.29	0.9322	7.1320	26.50	0.8182	5.4762	31.18	0.9119	6.8464

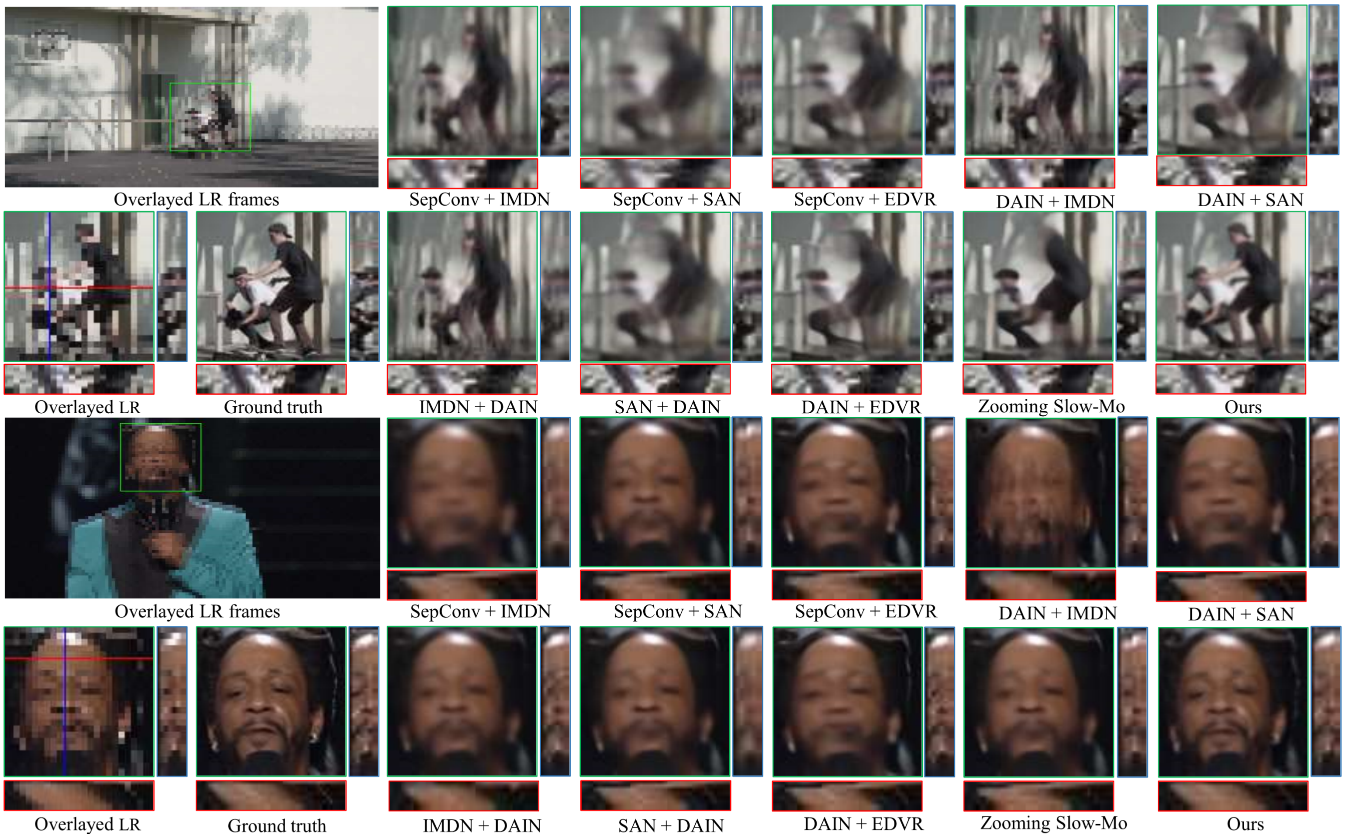

Overlayed LR frames
SepConv + IMDN
SepConv + SAN
SepConv + EDVR
DAIN + IMDN
DAIN + SAN
Overlayed LR
Ground truth
IMDN + DAIN
SAN + DAIN
DAIN + EDVR
Zooming Slow-Mo
Ours
Overlayed LR frames
SepConv + IMDN
SepConv + SAN
SepConv + EDVR
DAIN + IMDN
DAIN + SAN
Overlayed LR
Ground truth
IMDN + DAIN
SAN + DAIN
DAIN + EDVR
Zooming Slow-Mo
Ours

Figure 6: Visual comparisons of different methods on video frames from Vimeo90K dataset. To visualize the temporal consistency in 2D, we plot the transition of red horizontal and blue vertical scanlines over time (horizontal and vertical TPs).

can be seen from the accompanied vertical and horizontal TPs, other comparison methods incur obvious temporal discontinuity, while our method is able to reconstruct temporally consistent results.

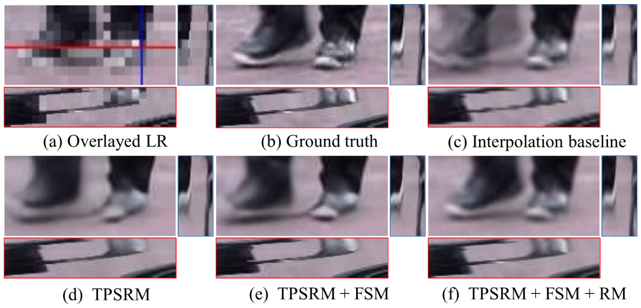
In Figure 9, we provide another observation of visual results in terms of error maps of the reconstructed video frames with respect to the ground truth. Note that the intermediate results of our method (*i.e.*, outputs from TPSRM and FSM) are also included for comparison. As can be seen, the output of RM is with less errors than the result of Zooming Slow-Mo, which demonstrates the effectiveness by exploiting the spatial-temporal correlation in the longer-term temporal context with TPs. On the other hand, we can see that the errors of outputs from the three modules of our method continue to decrease, which justifies the effectiveness of each module.

Table 2: Model parameters and average inference time on the Vid4 [20] testset with NVIDIA GTX1080Ti GPU.

Method	Parameters (Million)	Average Inference Time (s/frame)
DAIN [1] + EDVR [36]	24.0+20.7	0.8940
Zooming Slow-Mo [39]	11.10	0.1995
Ours	**7.53**	**0.1328**

Table 3: Investigation of different modules. Experimets are conducted on the Vid4 [20] testset.

Method	TPSRM	FSM	RM	PSNR↑	SSIM↑
(a)	✗	✗	✗	24.62	0.7626
(b)	✓	✗	✗	25.41	0.7743
(c)	✓	✓	✗	25.97	0.7976
(d)	✓	✓	✓	**26.50**	**0.8182**



(a) Overlayed LR (b) Ground truth (c) Interpolation baseline

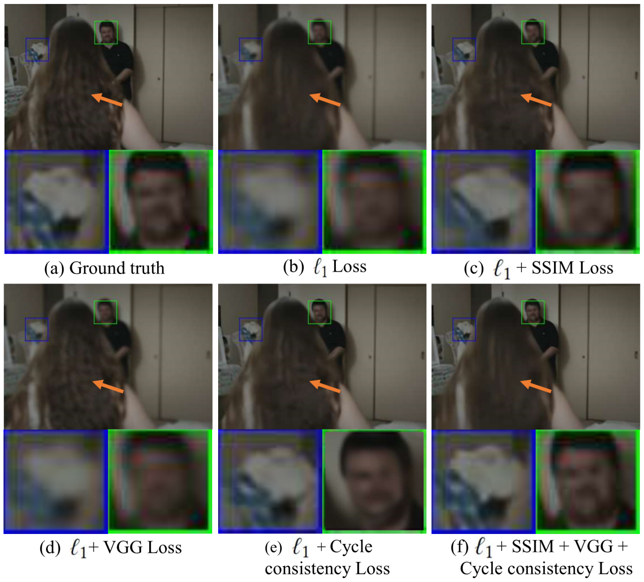
(d) TPSRM (e) TPSRM + FSM (f) TPSRM + FSM + RM

Figure 7: Visual comparisons of different modules. The frames are from the Vid4 [20] testset.

In Table 2, we investigate the model size and runtime of different networks on the Vid4 [20] testset with an NVIDIA GTX1080Ti GPU. Our model requires fewer parameters and less inference time than the typical two-stage and one-stage methods, which confirms that the proposed network is more lightweight and more efficient.

4.3 Ablation Study

Investigation of different modules. We conduct experiments to demonstrate the contributions of the three modules in our network. We use linear interpolation for VFI and bicubic interpolation for VSR as the baseline in our ablation study. The ablation results are shown in Table 3, with an exemplar visual comparison in Figure 7. Since TPSRM uses only vertical TPs and the spatial-temporal information has not been fully explored yet, the output of TPSRM are less detailed as shown in Figure 7. After adding FSM for spatial-temporal information fusion, the details become richer. But there still exists certain motion blur, which is further addressed after RM. By cascading all the three modules, we can obtain a continuous improvement in visual quality. The above observation is in accordance with the numerical results in Table 3.



(a) Ground truth (b) ℓ_1 Loss (c) ℓ_1 + SSIM Loss

(d) ℓ_1+ VGG Loss (e) ℓ_1 + Cycle consistency Loss (f) ℓ_1 + SSIM + VGG + Cycle consistency Loss

Figure 8: Visual comparisons of different combinations of losses. The frames are from the Vimeo90K-Slow [40] testset.

Table 4: Investigation of different losses. Experiments are conducted on the Vimeo90K-Slow [40] testset. The best and second results are highlighted in red and blue.

Loss functions setting	PSNR↑	SSIM↑	NIQE↓
ℓ_1	33.37	0.9177	7.1346
ℓ_1 + SSIM	33.39	0.9217	6.5150
ℓ_1 + VGG	33.37	0.9212	6.1064
ℓ_1 + Cycle	33.40	0.9216	6.4165
ℓ_1 + SSIM + VGG + Cycle	33.40	0.9217	6.1725

Investigation of different losses. We investigate the contribution of different loss terms by adjusting the weighting factors in Equation (10), and the results are shown in Table 4. When SSIM loss is adopted together with ℓ_1 loss, we can obtain the highest SSIM value. Since VGG loss is optimized at the feature level, the best result can be achieved in terms of the perceptual metric NIQE. Adding cycle consistency loss reinforces the consistency between the reconstructed frames and the LFR and LR inputs, which thus gives the highest PSNR value. Combining all of the four loss terms achieves an elegant overall performance. Figure 8 shows a visual comparison and it is clear that using only ℓ_1 loss produces over-smooth results. In contrast, the combination of ℓ_1 loss and SSIM loss helps improve the detailed structure of the reconstructed frame. The joint usage of ℓ_1 loss and VGG loss generates perception-oriented results, while using cycle consistency loss for network optimization avoids over-enhancement and alleviates possible artifacts. Based on the above observation, it is reasonable to use all the four loss terms to train our network.

4.4 Real-World Application: Old Movie Restoration

Due to the limited resolution of camera equipments, old movies often suffer from severe temporal and spatial degradations. In addition, preservation under different compression degrees further

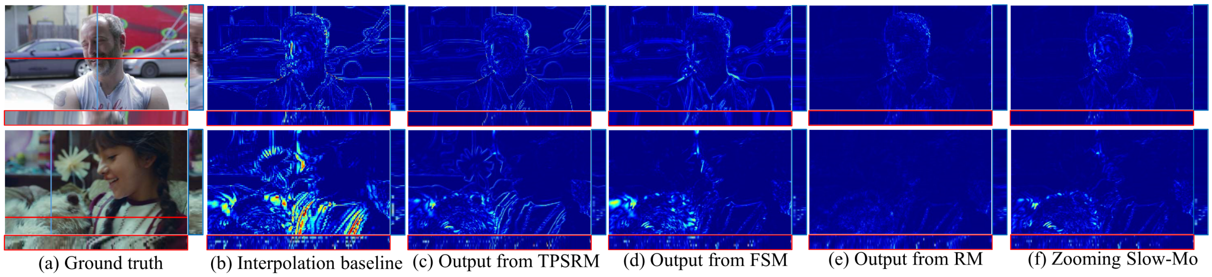


Figure 9: Visual results for different methods on the Vimeo90K [40] testset. Error maps show the residual between the output frame and the ground truth.

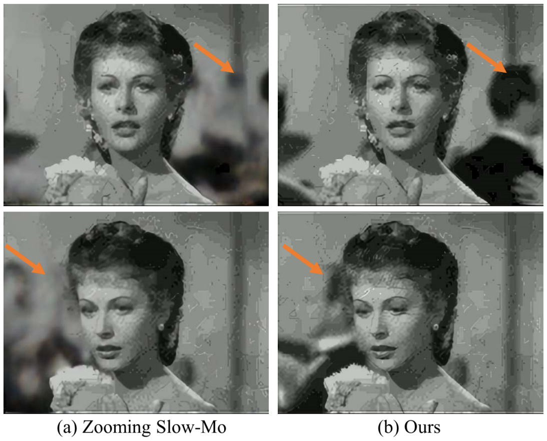


Figure 10: Qualitative evaluation on old movie restoration.

impacts the watching experience. Therefore, there is a large demand to convert these LFR and LR videos into HFR and HR ones with temporally-smooth and spatially-clear watching experience to satisfy the needs of modern displays.

In this section, we apply the proposed method along with Zooming Slow-Mo in the real-world scenario of old movie restoration. We download old movie clips from the Internet[3] as test data, and the network models are still trained on Vimeo90K [40]. Since no ground truth is available here, we can only evaluate the perceptual quality. As shown in Figure 10, both methods produce certain spatial artifacts. This is mainly because we do not explicitly address the compression issue in the network training, which is inconsistent with the real degradation process. Still, our method generates decent results in the background areas with movements, while Zooming Slow-Mo only generates spatially blurred results. It demonstrates that our model has better generalization ability.

4.5 Limitations

Despite of the promising performance as demonstrated above, the proposed method still has certain limitations in some challenging cases. For example, when a moving object suddenly appears or disappears in the video, it is difficult for the TPs to capture the global information due to the rapid movement. The failure case is shown in Figure 11. As the future work, we will explore the integration of horizontal and vertical TPs together to handle these complex movements in the STVSR task.

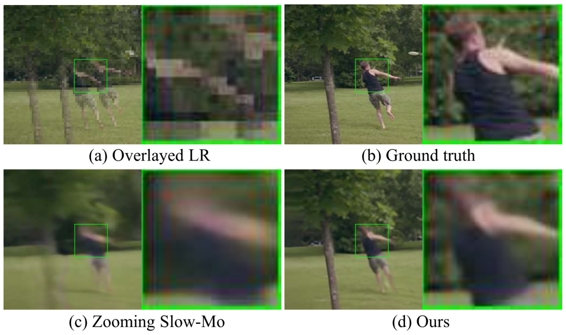


Figure 11: Failure case: when the moving object suddenly appears or disappears.

5 CONCLUSION

In this paper, we propose a novel one-stage method for space-time video super-resolution to up-convert video frame rate and generate high-resolution video frames simultaneously. The core contribution of this work is to introduce a new perspective, *i.e.*, temporal profiles, for exploitation of spatial-temporal information in videos. By using temporal profiles, the proposed network improves the efficiency and decrease the memory consumption without sacrificing the performance. While the temporal-profile super-resolution module directly captures the spatial-temporal correlation in the long-term temporal context, we then specifically design the feature shuffling module for spatial retargeting and spatial-temporal information fusion, following by the refining model for detail enhancement and artifacts alleviation. Comprehensive experimental results on a variety of testsets demonstrate that the proposed method achieves the new state-of-the-art performance for space-time video super-resolution.

ACKNOWLEDGMENTS

We acknowledge funding from National Key R&D Program of China under Grant 2017YFA0700800 and National Natural Science Foundation of China under Grants 61671419, 61901433, U19B2038 and 61620106009.

[3] https://tenor.com/

REFERENCES

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. 2019. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3703–3712.

[2] Robert C Bolles, H Harlyn Baker, and David H Marimont. 1987. Epipolar-plane image analysis: An approach to determining structure from motion. *International journal of computer vision* 1, 1 (1987), 7–55.

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4778–4787.

[4] Zhen Cheng, Zhiwei Xiong, Chang Chen, and Dong Liu. 2019. Light Field Super-Resolution: A Benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[5] Zhen Cheng, Zhiwei Xiong, and Dong Liu. 2019. Light Field Super-Resolution By Jointly Exploiting Internal and External Similarities. *IEEE Transactions on Circuits and Systems for Video Technology* 30 (2019), 2604–2616.

[6] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11065–11074.

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 2758–2766.

[8] Esmaeil Faramarzi, Dinesh Rajan, and Marc P Christensen. 2012. Space-time super-resolution from multiple-videos. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. IEEE, 23–28.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

[10] Alain Hore and Djemel Ziou. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*. IEEE, 2366–2369.

[11] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. 2019. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2024–2032.

[12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2462–2470.

[13] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. 2018. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9000–9008.

[14] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. 2018. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3224–3232.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

[16] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. 2016. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging* 2, 2 (2016), 109–122.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

[18] Tao Li, Xiaohai He, Qizhi Teng, Zhengyong Wang, and Chao Ren. 2015. Space–time super-resolution with patch group cuts prior. *Signal Processing: Image Communication* 30 (2015), 147–165.

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 136–144.

[20] Ce Liu and Deqing Sun. 2013. On Bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence* 36, 2 (2013), 346–360.

[21] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. 2017. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*. 4463–4471.

[22] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. 2012. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing* 21, 12 (2012), 4695–4708.

[23] Uma Mudenagudi, Subhashis Banerjee, and Prem Kumar Kalra. 2010. Space-time super-resolution using graph-cut optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5 (2010), 995–1008.

[24] Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 670–679.

[25] Simon Niklaus, Long Mai, and Feng Liu. 2017. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*. 261–270.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035.

[27] Jiayong Peng, Zhiwei Xiong, Yicheng Wang, Yueyi Zhang, and Dong Liu. 2020. Zero-Shot Depth Estimation From Light Field Using A Convolutional Neural Network. *IEEE Trans. Computational Imaging* 6 (2020), 682–696.

[28] Anurag Ranjan and Michael J Black. 2017. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4161–4170.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[30] Eli Shechtman, Yaron Caspi, and Michal Irani. 2002. Increasing space-time resolution in video. In *European Conference on Computer Vision*. Springer, 753–768.

[31] Eli Shechtman, Yaron Caspi, and Michal Irani. 2005. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 4 (2005), 531–545.

[32] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. 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*. 1874–1883.

[33] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[34] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. 2017. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*. 4472–4480.

[35] Joost van Amersfoort, Wenzhe Shi, Alejandro Acosta, Francisco Massa, Johannes Totz, Zehan Wang, and Jose Caballero. 2017. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *arXiv preprint arXiv:1711.06045* (2017).

[36] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. 2019. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[37] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[38] Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. 2017. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing* 11, 7 (2017), 926–954.

[39] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. 2020. Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[40] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision* 127, 8 (2019), 1106–1125.

[41] Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In *4th International Conference on Learning Representations, ICLR 2016*.

[42] Haochen Zhang, Dong Liu, and Zhiwei Xiong. 2019. Two-Stream Action Recognition-Oriented Video Super-Resolution. In *Proceedings of the IEEE International Conference on Computer Vision*. 8799–8808.