Joint Video Multi-Frame Interpolation and Deblurring under Unknown Exposure Time

Wei Shang[1], Dongwei Ren[1*], Yi Yang[1], Hongzhi Zhang[1], Kede Ma[2], Wangmeng Zuo[1,3]
[1]School of Computer Science and Technology, Harbin Institute of Technology
[2] City University of Hong Kong [3]Peng Cheng Laboratory, Shenzhen

Abstract

Natural videos captured by consumer cameras often suffer from low framerate and motion blur due to the combination of dynamic scene complexity, lens and sensor imperfection, and less than ideal exposure setting. As a result, computational methods that jointly perform video frame interpolation and deblurring begin to emerge with the unrealistic assumption that the exposure time is known and fixed. In this work, we aim ambitiously for a more realistic and challenging task - joint video multi-frame interpolation and deblurring under unknown exposure time. Toward this goal, we first adopt a variant of supervised contrastive learning to construct an exposure-aware representation from input blurred frames. We then train two U-Nets for intra-motion and inter-motion analysis, respectively, adapting to the learned exposure representation via gain tuning. We finally build our video reconstruction network upon the exposure and motion representation by progressive exposure-adaptive convolution and motion refinement. Extensive experiments on both simulated and real-world datasets show that our optimized method achieves notable performance gains over the state-of-the-art on the joint video $\times 8$ *interpolation and deblurring task. Moreover, on the seemingly implausible* $\times 16$ *interpolation task, our method outperforms existing methods by more than* 1.5 *dB in terms of PSNR.*

1. Introduction

When capturing videos, shutter period (*i.e.*, the inverse of the framerate) and exposure time are two major factors that we are able to manipulate for improved video quality, compared to other confounding factors such as object motion in the scene, lens imperfection, and sensor limitations [3]. Particularly, a long shutter period corresponds to lower framerate, and a longer exposure time increases the possibility of introducing severer motion blur (see Fig. 1). Often, we have good control over the shutter period in the form of the framerate, which is, nevertheless, quite limited in consumer cameras (*e.g.*, 30 frames per second, or FPS). This is not the case for exposure time, which may constantly and dynamically change depending on the video shooting environment, *e.g.*, illumination and reflection conditions [25]. Therefore, video frame interpolation (also known as framerate up-conversion [5, 10, 15]) and video deblurring methods (under unknown exposure time) are crucial for improving the quality of low framerate blurred videos, and are widely applicable to video editing, video compression, and slow-motion video generation.

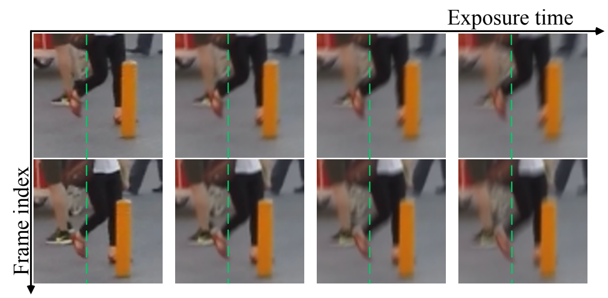

Figure 1. Visualization of motion blur under different exposure time and object motion. Frames from the left to right are obtained with increasing exposure time, leading to severer degree of motion blur. The situation may be worse if stronger motion is present, as illustrated by images in the second row.

In literature, video frame interpolation [21,30,42,48] and video deblurring [9, 35] have long been treated as individual problems and tackled separately with worth-celebrating successes. A straightforward approach to joint video frame interpolation and deblurring is to deploy deblurring methods followed by frame interpolation. However, this type of cascaded methods usually cannot obtain satisfactory reconstruction results, since algorithm-dependent deblurring artifacts would be propagated to and amplified in interpolated frames [41]. Similar situations will occur if we cascade video frame interpolation first [1]. This inspires recent work [1, 34, 41, 47] to cast video frame interpolation and deblurring (or super-resolution) as a joint and emerging low-level vision problem. However, these methods as-

*Corresponding author: rendongweihit@gmail.com.

sume known and fixed exposure time in the video degradation model, which is unrealistic, especially when the auto-exposure mode is enabled. Thus, they are bound to generalize poorly in the real-world.

Currently, there are two studies [25, 51] considering the setting of unknown exposure time. Zhang *et al*. [51] computed generic quadratic motion trajectories from consecutive blurred (or estimated sharp) frames. They directly cascaded existing deblurring and frame interpolation methods, suffering from the error propagation problem. Benefiting from additional event cameras [4], Kim [25] proposed an event-guided and end-to-end optimized video deblurring method under unknown exposure time, but event sensors have not been equipped on consumer cameras, restricting its practical applications.

In this work, we aim ambitiously for the more realistic and challenging task - joint video multi-frame interpolation and deblurring under unknown exposure time. Our primary design philosophy is *adaptive computation*: we adapt our Video frame Interpolation and Deblurring method under Unknown Exposure time (VIDUE) to exposure-aware and motion-aware representations, where the computation of the motion-aware representation is further adapted to the exposure-aware representation. To achieve this, we first adopt a variant of supervised contrastive learning to extract an exposure-aware representation from a low framerate blurred video. We then train two U-Nets: one is responsible for intra-motion analysis (*e.g*., assessing motion complexity within each frame); the other is responsible for inter-motion analysis (*e.g*., assessing motion continuity between frames). We adapt the two U-Nets to the exposure-aware representation via gain tuning [31] (also can be seen as a variant of sequence-and-excitation [16]). We last develop a video reconstruction network with a U-Net-like structure, which enables exposure-adaptive convolution and motion refinement in a progressive fashion.

Extensive experiments on both synthetic and real datasets show that the proposed VIDUE consistently produces higher-quality deblurred and $\times 8$ interpolated frames both visually and in terms of standard quality metrics. Moreover, VIDUE exhibits significant performance gains ($\geq$ 1.5 dB) over the state-of-the-art on the seemingly implausible $\times 16$ interpolation and deblurring task.

2. Related Work

2.1. Video Frame Interpolation

The video industry has long been interested in increasing the temporal resolution of a video sequence, *i.e*., framerate up conversion [7, 10, 18], with primary application to video compression. The problem resurges in the computer vision community under the name of video frame interpolation along with the renaissance of deep learning. Both Center-Frame Interpolation (CFI) [2, 11, 30, 33] and Multi-Frame Interpolation (MFI) [17, 19, 48] have been extensively investigated. For center-frame interpolation, Niklaus *et al*. [33] proposed separable kernel prediction networks to handle large motion, optimized by "perceptual" losses [20]. Bao *et al*. [2] proposed to incorporate depth information during interpolation to combat occlusion through bidirectional flow estimation. Lee *et al*. [30] combined and generalized the kernel-based and flow-based methods by offset prediction, and introduced an adversarial loss to examine the naturalness of the interpolated frame w.r.t. adjacent input frames. For multi-frame interpolation, more complex motion trajectory models need to be specified compared to the linear model assumption typically used in center-frame interpolation. Xu *et al*. [48] proposed a quadratic interpolation scheme to allow the inter-motion to be curvilinear. Chi *et al*. [8] proposed a cubic-based motion model with a relaxed warping loss to further boost interpolation accuracy for complex motion scenes. Huang *et al*. [17] designed the so-called privileged distillation scheme for real-time arbitrary timestamp frame interpolation. In addition, Kalluri *et al*. [21] cast multi-frame interpolation as a self-supervised pretext task to benefit downstream video applications, such as action recognition and video object tracking. 3D convolutions were adopted for spatiotemporal feature extraction. All these methods would encounter difficulties when processing motion-blurred videos because the optical flow/motion estimation as the core module will become less accurate.

2.2. Image and Video Deblurring

Traditionally, image deblurring is formulated as a Maximum A Posteriori (MAP) problem, which relies heavily on natural image priors, such as total variation [6], smoothness priors based on Markov random fields [37], normalized sparsity [28], and color-line priors [29]. Recent image deblurring methods adopt a pure data-driven approach, learning to deblur from massive deblurred-clean image pairs [9, 49]. Among popular architectural design choices, coarse-to-fine estimation has been extensively studied [9, 12, 32, 36]. For example, Chi *et al*. [9] adopted a U-Net to accept blurred images of multiple scales, and produce the corresponding set of sharp images in parallel. Alternatively, Ren *et al*. [38] proposed an unsupervised image deblurring scheme by leveraging deep priors [44] of both underlying sharp images and blur kernels. For video deblurring, the added temporal dimension can be coped with recurrent computation [14, 52], optical flow estimation [26, 35, 40], and deformable convolution [45], aided by handcrafted priors [35] or complementary modalities [40]. Image/video deblurring can only restore existing blurred frames, even motion information between input frames are computed to facilitate deblurring, which is kind of computational waste. Therefore, it is rational to consider joint video frame interpolation and deblurring, especially when the motion between consecutive frames is computed in either case.

2.3. Joint Video Interpolation and Deblurring

Recently, some researchers began to address the problem of joint video frame interpolation and deblurring. Shen *et al*. [41] proposed a pyramidal method for center-frame interpolation and deblurring. Argaw *et al*. [1] adopted a motion-based approach for not only multi-frame interpolation but also extrapolation. Oh *et al*. [34] introduced flow-guided attention and recursive feature refinement to improve the reconstruction performance. All these methods assume known and fixed exposure time, which is less realistic. Closest to ours, Zhang *et al*. [51] chose to cascade deblurring and interpolation networks under unknown exposure time. However, the performance would inevitably be compromised by the inaccuracy of the optical flow module in the deblurring network. Here, the proposed VIDUE is designed to adapt its computation to exposure-aware and motion-aware representations for joint video multi-frame interpolation and deblurring under unknown exposure time.

3. Proposed VIDUE

In this section, we first present the problem formulation of joint video frame interpolation and deblurring under unknown exposure time, and then describe in detail the proposed VIDUE method, consisting of an exposure-aware feature extractor g_e, an intra- and inter-motion analyzer g_a, and a video reconstruction network f.

3.1. Problem Formulation

When capturing a video frame, the shutter period Δt includes two phases: the exposure phase Δt_e and the effective readout phase Δt_r, where $\Delta t = \Delta t_e + \Delta t_r$. Given a video with T frames $\boldsymbol{y} = \{\boldsymbol{y}_t\}_{t=1}^{T}$, the t-th frame $\boldsymbol{y}_t$ is essentially an integral of the latent image $\boldsymbol{x}_\tau$ at each instant time τ over the exposure time Δt_e:

$$\boldsymbol{y}_t = \frac{1}{\Delta t_e}\int_{\tau=t\cdot\Delta t}^{\tau=t\cdot\Delta t+\Delta t_e} \boldsymbol{x}_\tau d\tau. \tag{1}$$

For consumer-grade cameras, the captured videos for dynamic scenes usually suffer from low framerate and blur due to long shutter period with a large portion of exposure time. Joint video $\times S$ interpolation and deblurring aims to recover a high framerate video with sharp frames: $\boldsymbol{x} = \{\boldsymbol{x}_t\}_{t=1}^{T\times S}$ from the observed video $\boldsymbol{y}$. In this work, we assume the shutter period Δt is known, while the exposure time Δt_e is not, and thus Δt_e is directly linked to the strength of the motion blur. It is noteworthy that the formulation in Eq. (1) is different from DeMFI [34], which synthesizes blur by directly averaging consecutive frames, while ignoring the readout phase.

3.2. Exposure-Aware Feature Extractor

Our first design choice is that the video reconstruction network f should adapt to different exposure time. For the $\times S$ interpolation task, we have the same number of S exposure time durations, where $\Delta t_e \in \{1, 2, \ldots, S\}$ and $\Delta t = S$ in Eq. (1). $\Delta t_e : \Delta t = S : S$ means that the effective readout time Δt_r is zero. That is, the actual readout phase is fully overlapped with the next exposure phase.

We work with a mini-batch of input video sequences, from which we create two multiviewed versions by applying horizontal and vertical flipping, 90° rotation, and random cropping to obtain $\mathcal{B} = \{(\boldsymbol{y}^{(i)}, \Delta t_e^{(i)})\}_{i=1}^{|\mathcal{B}|}$ with $\boldsymbol{y}^{(i)} \in \mathbb{R}^{(T\times 3)\times H\times W}$. We combine the temporal and channel dimensions into one to naïvely enable spatiotemporal analysis using 2D convolutions. We adopt a variant of ResNet18 as $g_e : \mathbb{R}^{(T\times3)\times H\times W} \mapsto \mathbb{R}^{C\times1}$, where we replace the classification head with two Fully-Connected (FC) layers with leaky ReLU nonlinearity in between, to extract the exposure-aware feature representation $\boldsymbol{u}^{(i)} \in \mathbb{R}^{C\times 1}$.

We want to make the feature representations $\{\boldsymbol{u}^{(i)}\}$ corresponding to different exposure time to be as discriminative as possible. Thus we resort to supervised contrastive learning [24], and introduce the relative weighting to indicate the difference in exposure time between each sample and the anchor:

$$\ell_{\text{ws}} = \sum_{\boldsymbol{u}\in\mathcal{B}} \frac{-1}{|\mathcal{P}|}\sum_{\boldsymbol{v}\in\mathcal{P}} \log \frac{\exp\left(\boldsymbol{u}^\intercal\boldsymbol{v}/\alpha\right)}{\sum\limits_{\boldsymbol{v}'\in\mathcal{B}\setminus\{\boldsymbol{u}\}} w(\boldsymbol{u},\boldsymbol{v}')\cdot\exp\left(\boldsymbol{u}^\intercal\boldsymbol{v}'/\alpha\right)}, \tag{2}$$

where $\boldsymbol{u}$ denotes the anchor, $\mathcal{P}$ contains positive samples that share the same exposure time with the anchor, and α is a fixed temperature parameter. The relative weighting can be straightforwardly computed by $w(\boldsymbol{u}^{(i)}, \boldsymbol{u}^{(j)}) = \left|\Delta t_e^{(i)} - \Delta t_e^{(j)}\right|$. We also try to formulate the exposure-aware feature representation learning as ordinal regression [13], but obtain worse final results (see Table 4). After training, the exposure-aware feature extractor g_e is fixed during the training of the motion analyzer and the final reconstruction network.

3.2.1 Intra- and Inter-Motion Analyzer

Our second design choice is that the video reconstruction network f should adapt to different motion patterns presented in dynamic scenes. We choose to analyze both intra-motion within each video frame, which is relevant to motion complexity (captured in a given exposure time period Δt_e), and inter-motion between frames, which is pertinent to motion continuity (captured in a given shutter period Δt). Our motion analyzer $g_a : \mathbb{R}^{(T\times3)\times H\times W} \mapsto \mathbb{R}^{T\times1\times H\times W} \times \mathbb{R}^{(S\times T)\times2\times H\times W}$, computes, from an input video sequence $\boldsymbol{y}^{(i)}$, T intra-motion maps and $S\times T$ inter-motion maps of the same spatial size, respectively, whose

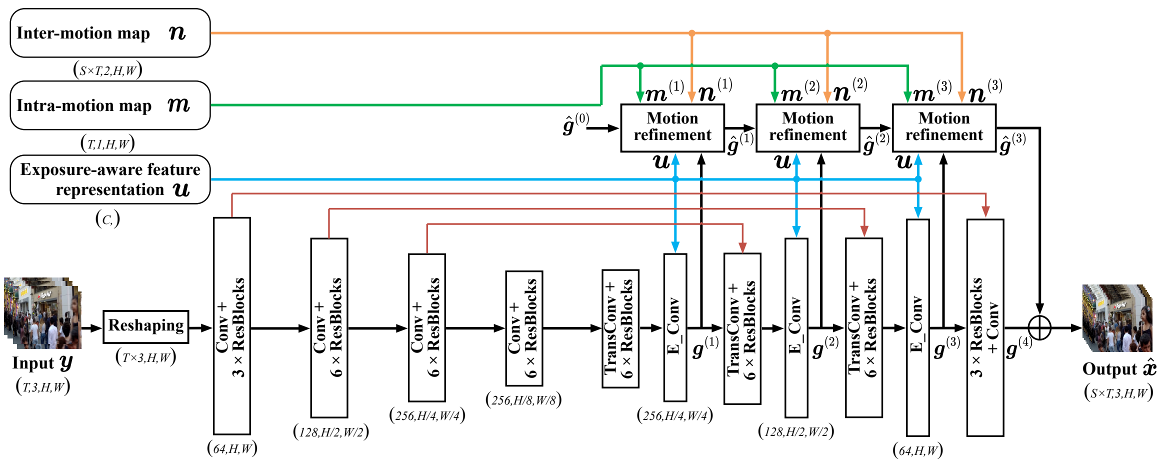

Figure 2. Overview of the reconstruction network of VIDUE, by which low framerate blurred video $\boldsymbol{y}$ with T frames can be reconstructed to high framerate sharp video with $S \times T$ frames for the $\times S$ interpolation and deblurring task. The reconstruction network is built upon the exposure-aware representation $\boldsymbol{u}$ and motion-aware representations $\boldsymbol{m}$ and $\boldsymbol{n}$ by exposure-adaptive convolution, and we implement progressive motion refinement for better reconstruction performance.

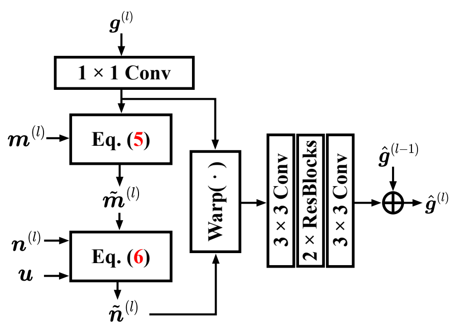

Figure 3. Overview of the proposed progressive motion refinement module. We first refine the intra-motion maps $\boldsymbol{m}^{(l)}$ with the help of $\boldsymbol{g}^{(l)}$ to obtain $\tilde{\boldsymbol{m}}^{(l)}$, and then refine the inter-motion maps $\boldsymbol{n}^{(l)}$ based on $\tilde{\boldsymbol{m}}^{(l)}$ and $\boldsymbol{u}$. Finally, we obtain the motion-refined output feature $\hat{\boldsymbol{g}}^{(l)}$ by warping the input feature $\boldsymbol{g}^{(l)}$ according to $\tilde{\boldsymbol{n}}^{(l)}$, followed by further refinement and incorporation of the upsampled $\hat{\boldsymbol{g}}^{(l-1)}$ from the previous stage via addition.

computation is adaptive to the exposure-aware representation $\boldsymbol{u}^{(i)}$.

Intra-Motion Analysis. We adopt a pre-trained light-weight U-Net [50], in which we tune the "gain" (*i.e*., a single multiplicative scaling parameter) of each channel of the intermediate representations $\boldsymbol{z} \in \mathbb{R}^{C' \times H' \times W'}$ (in the expansive path [39]):

$$\boldsymbol{g}_{i\bullet} = \boldsymbol{u}'_i \cdot \boldsymbol{z}_{i\bullet}, \quad \boldsymbol{u}' = \sigma\left(\mathbf{W}_2^\intercal \mathrm{LReLU}\left(\mathbf{W}_1^\intercal \boldsymbol{u}\right)\right). \qquad (3)$$

Here $\boldsymbol{z}_{i\bullet}$ stands for the i-th channel, and $\boldsymbol{u}' \in \mathbb{R}^{C' \times 1}$ is the gain vector computed from the exposure-aware representation $\boldsymbol{u}$ by two FC layers with leaky ReLU in between, followed by a Sigmoid function $\sigma(\cdot)$. $\{\mathbf{W}_1, \mathbf{W}_2\}$ are learnable weight matrices. Eq. (3) can also be seen as a form of the squeeze-and-excitation operation [16], where we "squeeze" the raw video sequence into $\boldsymbol{u}'$ through $\boldsymbol{u}$, and use it to "excite" $\boldsymbol{z}$. The results from the U-Net are intra-motion offsets $\boldsymbol{o}^{(s)} \in \mathbb{R}^{T \times 2 \times H \times W}$ and $\boldsymbol{o}^{(e)} \in \mathbb{R}^{T \times 2 \times H \times W}$, which are the starting and ending positions of estimated motion trajectories in horizontal and vertical directions, respectively. We compute the final intra-motion maps $\boldsymbol{m} \in \mathbb{R}^{T \times 1 \times H \times W}$ by subtracting $\boldsymbol{o}^{(e)}$ from $\boldsymbol{o}^{(s)}$, followed by root mean squared of the subtracted offsets. Intra-motion maps can also be integrated into g_e to improve representation capabilities.

Inter-Motion Analysis. We adopt a second randomly initialized light-weight U-Net, taking the estimated intra-motion offsets $\boldsymbol{o}^{(s)}$ and $\boldsymbol{o}^{(e)}$ as inputs, and producing inter-motion maps $\boldsymbol{n} \in \mathbb{R}^{(S \times T) \times 2 \times H \times W}$. The adaptive gain tuning in Eq. (3) is also enabled in the expansive path of the U-Net. The detailed network specifications of the motion analyzer can be found in the supplementary.

3.3. Video Reconstruction Network

It is then ready to describe our video reconstruction network $f : \mathbb{R}^{T \times 3 \times H \times W} \mapsto \mathbb{R}^{(S \times T) \times 3 \times H \times W}$, which is built upon the exposure-aware representation $\boldsymbol{u}$ and the motion-aware representations $\boldsymbol{m}$ and $\boldsymbol{n}$ by progressive exposure-adaptive convolution and motion refinement. The architecture is shown in Fig. 2.

We first feed the input video sequence $\boldsymbol{y} \in \mathbb{R}^{(T \times 3) \times H \times W}$ into an encoder, which consists of four stages of residual blocks, separated by the spatially downsampling layers. Each residual block contains two 3×3 convolution layers with leaky ReLU in between. Similarly, the decoder is composed of three stages of one transposed convolution for upsampling, residual blocks, one exposure-adaptive convolution, and one motion refinement module,

Method	Inference time (sec) / Framerate (FPS)	GoPro-5:8			GoPro-7:8		
		Deblurring	Interpolation	Avg	Deblurring	Interpolation	Avg
CDVD-TSP+QVI	1.02 / 7.84	33.06 / 0.952	30.02 / 0.914	30.40 / 0.918	31.39 / 0.933	29.39 / 0.902	29.64 / 0.906
CDVD-TSP+RIFE	0.40 / 20.00		29.95 / 0.907	30.34 / 0.913		29.15 / 0.892	29.44 / 0.897
MIMOUNetPlus+QVI	0.78 / 10.26	34.42 / 0.961	30.48 / 0.920	30.97 / 0.925	33.15 / 0.951	30.18 / 0.915	30.55 / 0.920
MIMOUNetPlus+RIFE	0.16 / 50.00		30.83 / 0.921	31.28 / 0.926		30.33 / 0.913	30.69 / 0.918
UTI-VFI	0.68 / 11.76	—	—	34.40 / 0.965	—	—	33.28 / 0.955
FLAVR	0.50 / 16.00	35.78 / 0.968	34.20 / 0.959	34.39 / 0.960	33.72 / 0.953	33.36 / 0.952	33.41 / 0.952
DeMFI	3.19 / 2.51	36.69 / 0.974	34.77 / 0.965	35.01 / 0.967	35.01 / 0.964	34.43 / 0.962	34.50 / 0.962
VIDUE (Ours)	0.27 / 29.63	**37.74 / 0.980**	**36.12 / 0.973**	**36.32 / 0.974**	**36.12 / 0.973**	**35.56 / 0.970**	**35.63 / 0.970**

Table 1. PSNR / SSIM comparison results on the GoPro dataset. "Deblurring" and "Interpolation" columns contain the reconstruction results of the input and interpolated frames, respectively, while the "Avg" column summarizes the average performance.

Figure 4. Visual comparison on the GoPro dataset. Orange and green boxes indicate cropped patches for zoomed-in comparison from deblurred and interpolated frames, respectively. TSP and MIMO are short for CDVD-TSP and MIMOUNetPlus, respectively. GT stands for the ground-truth patches cropped from the reference high framerate video.

followed by a back-end residual block and a convolution layer for channel number adjustment.

Exposure-Adaptive Convolution. Motivated by [22, 23], we propose to further convolve the intermediate representation $\boldsymbol{z}$ from the residual blocks using a filter bank, in which we adapt the filter weights to the exposure-aware representation $\boldsymbol{u}$. We generate the filter weights $\boldsymbol{w}$ by

$$\begin{aligned} \boldsymbol{u}' &= \mathbf{W}_2^\intercal \mathrm{LReLU}\left(\mathbf{W}_1^\intercal \boldsymbol{u}\right), \\ \boldsymbol{w}_{ijk} &= \frac{\boldsymbol{w}'_{ijk} \cdot \boldsymbol{u}'_i}{\sqrt{\sum_{i,k} (\boldsymbol{w}'_{ijk} \cdot \boldsymbol{u}'_i)^2 + \epsilon}}, \end{aligned} \tag{4}$$

where $\{\boldsymbol{w}', \mathbf{W}_1, \mathbf{W}_2\}$ are learnable, and i, j, and k index the input feature channel, the output feature channel, and the spatial footprint of the convolution, respectively. ϵ is a small positive constant to avoid numerical instability issues. As in [23], we first perform instance normalization of the feature representation $\boldsymbol{z}$ with the learnable scaling factor and the bias term, followed by the exposure-adaptive convolution to obtain $\boldsymbol{g}$ (refer to the notations in Fig. 2).

Motion Refinement Module. At the l-th stage for $l \in \{1, 2, 3\}$, the motion refinement module accepts $\boldsymbol{g}^{(l)}$ from the exposure-adaptive convolution as input, whose channel dimension is adjusted by a 1×1 convolution. The intra-motion maps $\boldsymbol{m}$ and the inter-motion maps $\boldsymbol{n}$ are accordingly resized to the same spatial size with $\boldsymbol{g}^{(l)}$, denoted by $\boldsymbol{m}^{(l)}$ and $\boldsymbol{n}^{(l)}$, respectively. We refine the intra-motion maps by

$$\tilde{\boldsymbol{m}}^{(l)} = \sigma\left(\mathrm{Conv}\left(\mathrm{Concat}\left(\boldsymbol{m}^{(l)}, \boldsymbol{g}^{(l)}\right)\right) + \boldsymbol{m}^{(l)}\right), \tag{5}$$

where $\mathrm{Conv}(\cdot)$ denotes a 3×3 convolution, $\mathrm{Concat}(\cdot)$ denotes the concatenation operation along the channel dimension, and $+$ indicates that we rely on a residual connection for implementation. We continue to refine the inter-motion maps $\boldsymbol{n}^{(l)}$ based on $\tilde{\boldsymbol{m}}^{(l)}$:

$$\tilde{\boldsymbol{n}}^{(l)} = \mathrm{Conv}(\mathrm{E\_Conv}(\boldsymbol{n}^{(l)}, \boldsymbol{u}) \odot \tilde{\boldsymbol{m}}^{(l)} + \boldsymbol{n}^{(l)} \odot (1 - \tilde{\boldsymbol{m}}^{(l)})), \tag{6}$$

where $\mathrm{E\_Conv}(\cdot)$ stands for the exposure-adaptive convolution and $\odot$ is an element-wise multiplication operation.

Progressive Reconstruction. Finally, we make use of the refined inter-motion maps $\tilde{\boldsymbol{n}}^{(l)}$ and the input features $\boldsymbol{g}^{(l)}$ for progressive reconstruction:

$$\hat{\boldsymbol{g}}^{(l)} = \mathrm{Refine}(\mathrm{Warp}(\boldsymbol{g}^{(l)}, \tilde{\boldsymbol{n}}^{(l)})) + \mathrm{Up}(\hat{\boldsymbol{g}}^{(l-1)}), \tag{7}$$

where $\mathrm{Warp}(\cdot)$ denotes the backward warping function [35], and $\mathrm{Up}(\cdot)$ denotes the bilinear upsampling operation. $\mathrm{Refine}(\cdot)$ is implemented by a front-end convolution layer, two residual blocks, and a back-end convolution layer. To initialize progressive reconstruction, we set $\hat{\boldsymbol{g}}^{(0)} = \mathbf{0}$ as a tensor with all zeros, and summarize one stage of processing in Fig. 3. At last, we add $\boldsymbol{g}^{(4)}$ (*i.e.*, the output of the back-end residual block and convolution layer in the decoder) to $\hat{\boldsymbol{g}}^{(3)}$ to estimate the high framerate sharp video sequence $\hat{\boldsymbol{x}} \in \mathbb{R}^{(S \times T) \times 3 \times H \times W}$.

During training, we optimize all modules in the proposed VIDUE method (except for the exposure-aware feature extractor) using a variant of stochastic gradient descent by minimizing the Mean Absolute Error (MAE) between the ground-truth high framerate sharp video sequences and their predictions by VIDUE.

Method	Adobe-5:8			Adobe-7:8		
	Deblurring	Interpolation	Avg	Deblurring	Interpolation	Avg
CDVD-TSP+QVI	29.34 / 0.896	23.34 / 0.739	24.10 / 0.759	27.57 / 0.862	23.42 / 0.741	23.94 / 0.756
CDVD-TSP+RIFE		23.29 / 0.729	24.06 / 0.750		23.31 / 0.730	23.85 / 0.747
MIMOUNetPlus+QVI	29.71 / 0.894	23.48 / 0.741	24.27 / 0.761	27.69 / 0.854	23.52 / 0.740	24.05 / 0.754
MIMOUNetPlus+RIFE		23.85 / 0.748	24.59 / 0.767		23.76 / 0.744	24.26 / 0.758
UTI-VFI	—	—	26.69 / 0.842	—	—	25.84 / 0.815
FLAVR	28.52 / 0.869	27.05 / 0.841	27.23 / 0.845	26.98 / 0.834	26.94 / 0.837	26.94 / 0.837
DeMFI	27.24 / 0.848	25.49 / 0.814	25.71 / 0.818	25.81 / 0.813	25.64 / 0.814	25.66 / 0.814
VIDUE (Ours)	**30.44 / 0.905**	**28.50 / 0.876**	**28.74 / 0.880**	**27.85 / 0.864**	**27.92 / 0.867**	**27.91 / 0.866**

Table 2. PSNR / SSIM comparison results on the Adobe dataset.

Figure 5. Visual comparison on the Adobe dataset. Orange and green patches are from deblurred and interpolated frames, respectively.

4. Experiments

In this section, we evaluate VIDUE on both synthetic and real-world datasets. More results including reconstructed videos can be found in the supplementary. The source code is implemented in Pytorch, and is made publicly available at https://github.com/shangwei5/VIDUE. We also provide an implementation in HUAWEI Mindspore at https://github.com/Hunter-Will/VIDUE-mindspore.

4.1. Datasets

To establish simulated datasets for quantitative performance comparison of video interpolation and deblurring, we synthesize low framerate videos according to Eq. (1) by downsampling high framerate videos [34, 51]. The experiments are conducted on both the GoPro dataset [32] and the Adobe dataset [43]. On the GoPro dataset with $\times 8$ interpolation (and deblurring) task, we set $S = 8$ by which an original 240 FPS video is degraded to 30 FPS. For a fair comparison, we set exposure frames to two odd numbers, *i.e.*, $\Delta t_e : \Delta t \in \{5 : 8, 7 : 8\}$, since existing methods require the middle frame as reference. As for training, we sample $\Delta t_e : \Delta t \in \{1 : 8, 2 : 8, \ldots, 8 : 8\}$ to generate blurred frames as a form of data augmentation to train the exposure-aware feature extractor g_e. On the Adobe dataset with $\times 8$ interpolation (and deblurring) task, the synthetic setting is identical, but the blurring artifacts are more severe by setting a longer shutter period. Furthermore, we evaluate the generalizability of VIDUE against competing methods on real-world data from the RealBlur dataset [43].

4.2. Implementation Details

We set the input frame number $T = 4$, and the the exposure-aware feature dimension $C = 256$. The temperature parameter in Eq. (2) and the normalizing constant in Eq. (4) is set to $\alpha = 0.5$ and $\epsilon = 1 \times 10^{-5}$, respectively. We adopt the Adam [27] optimizer with the default setting for training g_e with an initial learning rate of 0.1 and a mini-batch size of 40. Similarly, we adopt the AdaMax [27] optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for training g_a and f, and set the initial learning rate and the mini-batch size to 2×10^{-4} and 12, respectively. We train the models for 200 epochs, and halve the learning rate whenever the training plateaus, which is cross-validated as done in [21]. VIDUE is trained on 4 Tesla V100 GPUs, and can make inference on a single GTX 2080 Ti GPU. Throughout the paper, we use the Peak Signal-to-Noise Ratio (PSNR) and the Structural SIMilarity (SSIM) index [46] as the evaluation metrics.

4.3. Evaluation on $\times 8$ Interpolation Task

We compare VIDUE with both cascade and joint methods. For cascade methods, deblurring methods CDVD-TSP [35], MIMOUNetPlus [9] and interpolation methods QVI [48], RIFE [17] are cascaded for video deblurring and interpolation. We also take UTI-VFI [51] into comparison, which handles unknown exposure time. For joint methods, we include FLAVR [21] and DeMFI [34]. It is worth noting that the original FLAVR is an interpolation method with sharp inputs, and we retrain it on the same training sets to tackle joint video interpolation and deblurring. For MIMOUNetPlus and RIFE, we find that models provided by the respective authors perform better than our retrained counterparts, and thus we stick to the official implementations for evaluation.

Method	GoPro-9:16	GoPro-11:16	GoPro-13:16	GoPro-15:16	Avg
CDVD-TSP+QVI	25.70 / 0.786	25.45 / 0.780	25.20 / 0.774	24.97 / 0.767	25.33 / 0.777
CDVD-TSP+RIFE	26.08 / 0.797	25.74 / 0.788	25.41 / 0.780	25.12 / 0.772	25.59 / 0.784
MIMOUNetPlus+QVI	26.12 / 0.794	25.97 / 0.791	25.78 / 0.787	25.49 / 0.780	25.84 / 0.788
MIMOUNetPlus+RIFE	26.95 / 0.821	26.71 / 0.815	26.41 / 0.808	25.98 / 0.797	26.51 / 0.810
UTI-VFI	28.59 / 0.876	28.17 / 0.866	27.40 / 0.845	26.37 / 0.814	27.63 / 0.851
FLAVR	28.77 / 0.875	28.71 / 0.875	28.43 / 0.870	28.04 / 0.860	28.49 / 0.870
DeMFI	28.64 / 0.876	28.67 / 0.878	28.49 / 0.873	28.11 / 0.863	28.48 / 0.873
VIDUE (Ours)	**31.10 / 0.922**	**29.91 / 0.906**	**29.90 / 0.905**	**29.50 / 0.898**	**30.10 / 0.908**

Table 3. PSNR / SSIM comparison results on the GoPro dataset for joint $\times 16$ interpolation and deblurring.

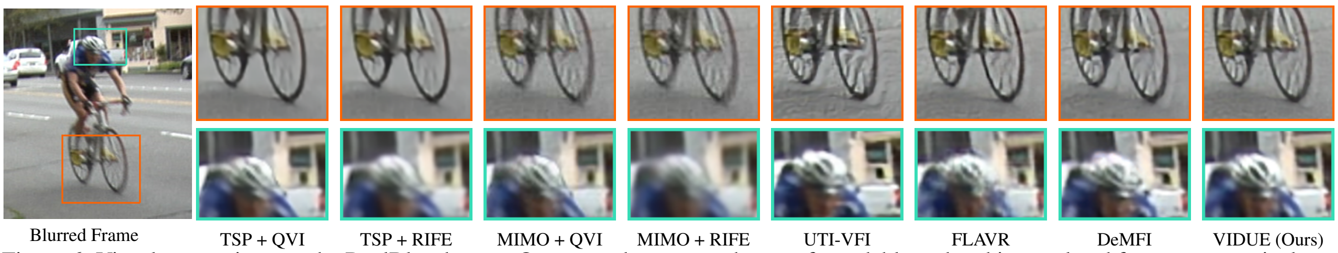

Figure 6. Visual comparison on the RealBlur dataset. Orange and green patches are from deblurred and interpolated frames, respectively.

Setting	GoPro-5:8	GoPro-7:8
Exposure-Agnostic	34.79 / 0.965	34.13 / 0.960
Ordinal Regression	35.44 / 0.969	34.97 / 0.966
Contrastive Learning	36.32 / 0.974	35.63 / 0.970
Known Exposure Time	36.41 / 0.974	35.87 / 0.971

Table 4. Role of the exposure-aware feature representation $\boldsymbol{u}$ evaluated by PSNR / SSIM. The default setting is highlighted in bold.

4.3.1 Comparison on the GoPro Dataset

As mentioned previously, we create two versions of the GoPro dataset, "GoPro-5:8" and "GoPro-7:8", with different exposure time, and list the comparison results in Table 1. We find that cascade methods are significantly worse than joint methods, with a PSNR gap of almost 3 to 6 dB. In comparison with joint methods, VIDUE achieves about 1.0 to 1.1 dB PSNR gains for deblurring, and about 1.1 to 1.4 dB PSNR gains for interpolation, respectively. In terms of visual comparison in Fig. 4, cascade methods fail in interpolating sharp frames. The deblurring results of the joint methods FLAVR and DeMFI are satisfactory, but they are still limited in interpolation. This is because the unknown exposure time setting is not carefully modeled, and latent sharp frames during the readout phase cannot be properly reconstructed. In stark contrast, VIDUE is able to adapt to different exposure time based on the learned exposure-aware feature representation, and achieves noticeable performance gains on this challenging task of joint video multi-frame interpolation and deblurring. In addition, we test the inference time (and framerate) of all methods on the GoPro dataset. We calculate the average running time of reconstructing 8 frames (*i.e.*, using $\times 8$ interpolation as the example) on the Tesla V100 GPU. We find from Table 1 that VIDUE shows clear advantages over all competing methods except for the light-weight cascade method (*i.e.*, MIMOUNetPlus+RIFE, which does not deliver convincing reconstruction performance). DeMFI involves extensive recurrent computation, leading to the longest inference time among all methods. In summary, the proposed VIDUE enjoys faster inference time, and achieves the best reconstruction performance, which justifies our key design philosophy of adaptive computation to deblurring and interpolation relevant features.

4.3.2 Comparison on the Adobe Dataset

We evaluate the joint $\times 8$ interpolation and deblurring performance of VIDUE against the competing methods in Table 2 and Fig. 5. Despite more severe blurring artifacts than those in the GoPro dataset, we come to similar conclusions that joint methods are superior over cascade methods. The most competitive method - DeMFI - experiences a significant performance drop due to the presence of the heavy blur. VIDUE still outperforms FLAVR about 0.9 to 1.9 dB for deblurring, and about 1.0 to 1.4 dB for interpolation, respectively. From Fig. 5, one can see that the competing methods fail to restore sharp frames, while VIDUE still obtains visually favorable results even in the presence of the strong blur.

4.3.3 Generalization to Real-World Videos

Finally, we evaluate the generalizability of VIDUE on real-world blurred frames by testing the model trained on the GoPro dataset to interpolate and deblur video data from the RealBlur dataset. As shown in Fig. 6, VIDUE achieves the most visually plausible interpolation and deblurring results with sharper structures and textures, while others suffer from visually annoying artifacts.

Exposure Representation		Intra-Motion	Inter-Motion	Motion Refinement	GoPro-5:8	GoPro-7:8
Squeeze-and-Excitation	E_Conv					
✗	✗	✗	✗	✗	34.79 / 0.965	34.13 / 0.960
✗	✓	✗	✗	✗	36.05 / 0.972	34.87 / 0.963
✗	✓	✗	✓	✗	36.18 / 0.973	35.27 / 0.968
✗	✓	✓	✗	✓	36.20 / 0.973	35.24 / 0.968
✗	✓	✓	✓	✗	36.23 / **0.974**	35.28 / 0.968
✓	✗	✓	✓	✓	36.12 / 0.973	35.39 / 0.969
✗	✓	✓	✓	✓	**36.32 / 0.974**	**35.63 / 0.970**

Table 5. Module analysis of VIDUE on the GoPro dataset.

4.4. Evaluation on $\times 16$ Interpolation Task

Without auxiliary information (*e.g.*, event signals [25]), joint $\times 16$ interpolation and deblurring is extremely difficult, and is rarely evaluated in literature. We conduct this experiment on the GoPro dataset, and list the results in Table 3. VIDUE achieves more than 1.5 dB gains than existing methods, and performs consistently better under all exposure time settings, which are unknown. More visual comparisons are provided in the supplementary.

4.5. Ablation Studies

4.5.1 Effectiveness of Exposure-Aware Representation

As shown in Table 4, four VIDUE variants are trained on the GoPro dataset to verify the effectiveness of the exposure-aware feature representation $\boldsymbol{u}$: 1) one that is exposure-agnostic (*i.e.*, without learning and adapting to $\boldsymbol{u}$), 2) one that learns $\boldsymbol{u}$ using ordinal regression, 3) one that learns $\boldsymbol{u}$ using supervised contrastive learning combined with relative weighting (as the default setting), 4) one with the known exposure time (as a form of upper bound). To leverage the known exposure time, we first represent it as an S-dimensional vector with the first Δt_e entries being one and the remaining entries being zero. We then use two FC layers with leaky ReLU in between to map it into a C-dimensional feature representation, which can be readily adapted in VIDUE. As reported in Table 4, both exposure-aware feature representations learned by ordinal regression and supervised contrastive learning bring significant improvements than the exposure-agnostic variant. Compared to ordinal regression, our default choice of supervised contrastive learning is able to approach the "upper bound" with the known exposure time. We believe this arises because contrastive learning provides a more direct way of encouraging discriminative feature learning than ordinal regression learns to rank different exposure time.

4.5.2 Module Analysis

We single out the contribution of each component of VIDUE using the GoPro dataset. The first row in Table 5 shows the results of a plain U-Net, while the second row is obtained by a simplified VIDUE without motion analysis and adaptation in the decoder. Surprisingly, the simplified VIDUE achieves better results than existing methods, (see Tables 1 and 5). The performance gains by VIDUE is mostly attributed to the use of the exposure-aware representation by contrasting it to the plain U-Net. We next remove each component of VIDUE to verify its necessity. Most importantly, removing motion refinement leads to performance drops under different exposure time, especially when the exposure time is large (*i.e.*, when the motion is strong). We also replace the exposure-adaptive convolution (*i.e.*, E_Conv defined in Eq. (4)) with the sequence-and-excitation operation [16], and find that E_Conv is more effective in leveraging the exposure-aware representation. As expected, the full VIDUE achieves the best interpolation and deblurring performance.

5. Conclusion

We have described a computational method - VIDUE - for joint video multi-frame interpolation and deblurring under unknown exposure time. We trained contrastively to extract exposure-aware feature representation, which can then be embedded into intra- and inter-motion analyzer and the video reconstruction network via gain tuning and exposure-adaptive convolution, respectively. We refined the estimated motion representations for better progressive video reconstruction. We demonstrated the superiority of VIDUE on both synthetic and real-world datasets to perform $\times 8$ and $\times 16$ interpolation and deblurring tasks. Future work can be planed to 1) further reduce the computational complexity of VIDUE while retaining (or improving) the performance and 2) to optimize VIDUE by perceptual quality metrics with emphasis on temporal coherence.

Acknowledgements

This work was supported in part by the National Key Research and Development Project (2022YFA1004100), the National Natural Science Foundation of China (62172127, 62071407, and U22B2035), the Natural Science Foundation of Heilongjiang Province (YQ2022F004), the Hong Kong RGC Early Career Scheme (9048212), and the CAAI-Huawei MindSpore Open Fund.

References

[1] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, and In So Kweon. Motion-blurred video interpolation and extrapolation. In *AAAI*, pages 901–910, 2021. 1, 3

[2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *CVPR*, pages 3703–3712, 2019. 2

[3] Alan C Bovik. *Handbook of Image and Video Processing*. Academic Press, 2010. 1

[4] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE JSSC*, 49:2333–2341, 2014. 2

[5] Roberto Castagno, Petri Haavisto, and Giovanni Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE TCSVT*, 6:436–446, 1996. 1

[6] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE TIP*, 7:370–375, 1998. 2

[7] Yen-Kuang Chen, Anthony Vetro, Huifang Sun, and Sun-Yuan Kung. Frame-rate up-conversion using transmitted true motion vectors. In *MSPW*, pages 622–627, 1998. 2

[8] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng Liu, Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *ECCV*, pages 107–123, 2020. 2

[9] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, pages 4641–4650, 2021. 1, 2, 6

[10] Byung-Tae Choi, Sung-Hee Lee, and Sung-Jea Ko. New frame rate up-conversion using bi-directional motion estimation. *IEEE TCE*, 46:603–609, 2000. 1, 2

[11] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, pages 10663–10671, 2020. 2

[12] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, pages 3848–3856, 2019. 2

[13] Pedro A Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. Ordinal regression methods: Survey and experimental study. *IEEE TKDE*, 28:127–146, 2015. 3

[14] Tae H Kim, Kyoung M Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *ICCV*, pages 4038–4047, 2017. 2

[15] Petri Haavisto, Janne Juhola, and Yrjö Neuvo. Fractional frame rate up-conversion using weighted median filters. *IEEE TCE*, 35:272–278, 1989. 1

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2, 4, 8

[17] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. RIFE: Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, pages 624–642, 2022. 2, 6

[18] Bo-Won Jeon, Gun-Ill Lee, Sung-Hee Lee, and Rae-Hong Park. Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. *IEEE TCE*, 49:499–508, 2003. 2

[19] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018. 2

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016. 2

[21] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. FLAVR: Flow-agnostic video representations for fast frame interpolation. In *WACV*, pages 2071–2082, 2023. 1, 2, 6

[22] Di Kang, Debarun Dhar, and Antoni Chan. Incorporating side information by adaptive convolution. In *NeurIPS*, pages 1–10, 2017. 5

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 5

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, pages 18661–18673, 2020. 3

[25] Taewoo Kim, Jeongmin Lee, Lin Wang, and Kuk-Jin Yoon. Event-guided deblurring of unknown exposure time videos. In *ECCV*, pages 1–34, 2022. 1, 2, 8

[26] Tae H Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, pages 106–122, 2018. 2

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[28] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 233–240, 2011. 2

[29] Wei-Sheng Lai, Jian-Jiun Ding, Yen-Yu Lin, and Yung-Yu Chuang. Blur kernel estimation using normalized color-line prior. In *CVPR*, pages 64–72, 2015. 2

[30] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. AdaCof: Adaptive collaboration of flows for video frame interpolation. In *CVPR*, pages 5316–5325, 2020. 1, 2

[31] Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter Crozier, Carlos Fernandez-Granda, and Eero P Simoncelli. Adaptive denoising via gaintuning. In *NeurIPS*, pages 23727–23740, 2021. 2

[32] Seungjun Nah, Tae H Kim, and Kyoung M Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 2, 6

[33] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, pages 261–270, 2017. 2

[34] Jihyong Oh and Munchurl Kim. DeMFI: Deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting. In *ECCV*, pages 1–18, 2022. 1, 3, 6

[35] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *CVPR*, pages 3043–3051, 2020. 1, 2, 5, 6

[36] Dongwon Park, Dong U Kang, Jisoo Kim, and Se Y Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *ECCV*, pages 327–343, 2020. 2

[37] Ashish Raj and Ramin Zabih. A graph cut algorithm for generalized image deconvolution. In *ICCV*, pages 1048–1054, 2005. 2

[38] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *CVPR*, pages 3341–3350, 2020. 2

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 4

[40] Wei Shang, Dongwei Ren, Dongqing Zou, Jimmy S Ren, Ping Luo, and Wangmeng Zuo. Bringing events into video deblurring with non-consecutively blurry frames. In *ICCV*, pages 4531–4540, 2021. 2

[41] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *CVPR*, pages 5114–5123, 2020. 1, 3

[42] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. XVFI: Extreme video frame interpolation. In *ICCV*, pages 14489–14498, 2021. 1

[43] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. 6

[44] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018. 2

[45] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen C Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, pages 0–0, 2019. 2

[46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13:600–612, 2004. 6

[47] Gang Xu, Jun Xu, Zhen Li, Liang Wang, Xing Sun, and Ming-Ming Cheng. Temporal modulation network for controllable space-time video super-resolution. In *CVPR*, pages 6388–6397, 2021. 1

[48] Xiangyu Xu, Siyao Li, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, pages 1–10, 2019. 1, 2, 6

[49] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, pages 5978–5986, 2019. 2

[50] Youjian Zhang, Chaoyue Wang, Stephen J Maybank, and Dacheng Tao. Exposure trajectory recovery from motion blur. *IEEE TPAMI*, 44:7490–7504, 2021. 4

[51] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. Video frame interpolation without temporal priors. In *NeurIPS*, pages 13308–13318, 2020. 2, 3, 6

[52] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, pages 191–207, 2020. 2