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

Wei Shang\textsuperscript{1}, Dongwei Ren\textsuperscript{1,*}, Yi Yang\textsuperscript{1}, Hongzhi Zhang\textsuperscript{1}, Kede Ma\textsuperscript{2}, Wangmeng Zuo\textsuperscript{1,3}

\textsuperscript{1}School of Computer Science and Technology, Harbin Institute of Technology
\textsuperscript{2}City University of Hong Kong
\textsuperscript{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 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 severe 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.

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

Currently, there are two studies [25, 51] considering the
setting of unknown exposure time. Zhang et al. [51] com-
puted generic quadratic motion trajectories from consec-
tutive blurred (or estimated sharp) frames. They directly
cascaded existing deblurring and frame interpolation meth-
ods, suffering from the error propagation problem. Benefi-
ting 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 interpola-
tion and deblurring under unknown exposure time. Our pri-
mary design philosophy is adaptive computation: we adapt
our Video frame Interpolation and Deblurring method un-
der 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 ex-
tract 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 representa-
tion via gain tuning [31] (also can be seen as a variant of
sequence-and-excitation [16]). We last develop a video re-
construction network with a U-Net-like structure, which en-
ables exposure-adaptive convolution and motion refinement
in a progressive fashion.

Extensive experiments on both synthetic and real
datasets show that the proposed VIDUE consistently pro-
duces 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 im-
plausible \( \times 16 \) interpolation and deblurring task.

2. Related Work

2.1. Video Frame Interpolation

The video industry has long been interested in increas-
ing the temporal resolution of a video sequence, i.e., fra-
frame conversion [7, 10, 18], with primary application to
video compression. The problem resurges in the computer
vision community under the name of video frame interpo-
lation 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 exten-
sively 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 bidirec-
tional flow estimation. Lee et al. [30] combined and gen-
eralized 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 in-
put 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 inter-
polation 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] de-
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 applica-
tions, such as action recognition and video object track-
ing. 3D convolutions were adopted for spatiotemporal fea-
ture extraction. All these methods would encounter diffi-
culties when processing motion-blurred videos because the
optical flow/motion estimation as the core module will be-
come less accurate.

2.2. Image and Video Deblurring

Traditionally, image deblurring is formulated as a Max-
umum A Posteriori (MAP) problem, which relies heav-
ily 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]. Re-
cent image deblurring methods adopt a pure data-driven
approach, learning to deblur from massive deblurred-clean
image pairs [9, 49]. Among popular architectural de-
sign 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 im-
age 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 estima-
tion [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 computa-
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 \( \Delta t \) includes two phases: the exposure phase \( \Delta t_e \) and the effective readout phase \( \Delta t_r \), where \( \Delta t = \Delta t_e + \Delta t_r \). Given a video with \( T \) frames \( y = \{y_t\}_{t=1}^T \), the \( t \)-th frame \( y_t \) is essentially an integral of the latent image \( x_t \) at each instant time \( \tau \) over the exposure time \( \Delta t_e \):

\[
y_t = \frac{1}{\Delta t_e} \int_{\tau = t \cdot \Delta t}^{\tau = (t+1) \cdot \Delta t} x_\tau d\tau.
\]

For consumer-grade cameras, the captured videos for dynamic scenes usually suffer from low framerate and blur due to long shutter periods with a large portion of exposure time. Joint video \( \times S \) interpolation and deblurring aims to recover a high framerate video with sharp frames: \( x = \{x_t\}_{t=1}^{T \times S} \) from the observed video \( y \). In this work, we assume the shutter period \( \Delta t \) is known, while the exposure time \( \Delta t_e \) is not, and thus \( \Delta t_e \) 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). \( \Delta t_e : \Delta t = S : \) means that the effective readout time \( \Delta t_e \) 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 \( B = \{(y^{(i)}, \Delta t^{(i)}_e)\}_{i=1}^{|B|} \) with \( y^{(i)} \in \mathbb{R}^{(T \times 3) \times H \times W} \). We combine the temporal and channel dimensions into one to naïvely enable spatiotemporal analysis using 2D convolutions. We adopt a variant of ResNet18 as \( g_e : \mathbb{R}^{(T \times 3) \times H \times W} \rightarrow \mathbb{R}^{C \times 1} \), 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 \( u^{(i)} \in \mathbb{R}^{C \times 1} \).

We want to make the feature representations \( \{u^{(i)}\} \) 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:

\[
\ell_{ws} = \sum_{u \in B} \frac{-1}{|P|} \sum_{u \in P} \log \frac{\exp (u^\top v / \alpha)}{\sum_{v' \in B \backslash \{u\}} \exp (u^\top v' / \alpha)},
\]

where \( u \) denotes the anchor, \( P \) contains positive samples that share the same exposure time with the anchor, and \( \alpha \) is a fixed temperature parameter. The relative weighting can be straightforwardly computed by \( w(u^{(i)}, u^{(j)}) = \left|\Delta t^{(i)}_e - \Delta t^{(j)}_e\right|\). 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 \( \Delta t_e \)), and inter-motion between frames, which is pertinent to motion continuity (captured in a given shutter period \( \Delta t \)). Our motion analyzer \( g_a : \mathbb{R}^{(T \times 3) \times H \times W} \rightarrow \mathbb{R}^{T \times 1 \times H \times W} \times \mathbb{R}^{(S \times T) \times 2 \times H \times W} \), computes, from an input video sequence \( y^{(i)} \), \( T \) intra-motion maps and \( S \times T \) inter-motion maps of the same spatial size, respectively, whose
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 \( z \in \mathbb{R}^{C' \times H' \times W'} \) (in the expansive path \([39]\)):

\[
g_i = u'_i \cdot z_i, \quad u' = \sigma(W_2^L \text{ReLU}(W_1^L u)). \quad (3)
\]

Here \( z_i \) stands for the \( i \)-th channel, and \( u' \in \mathbb{R}^{C' \times 1} \) is the gain vector computed from the exposure-aware representation \( u \) by two FC layers with leaky ReLU in between, followed by a Sigmoid function \( \sigma(\cdot) \). \( \{W_1, W_2\} \) 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 \( u' \) through \( u \), and use it to “excite” \( z \). The results from the U-Net are intra-motion offsets \( o^{(s)} \in \mathbb{R}^{T \times 2 \times H \times W} \) and \( o^{(c)} \in \mathbb{R}^{T \times 2 \times H \times W} \), which are the starting and ending positions of estimated motion trajectories in horizontal and vertical directions, respectively. We compute the final intra-motion maps \( m \in \mathbb{R}^{T \times 1 \times H \times W} \) by subtracting \( o^{(c)} \) from \( o^{(s)} \), 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 \( o^{(s)} \) and \( o^{(c)} \) as inputs, and producing inter-motion maps \( n \in \mathbb{R}^{(S \times T) \times 2 \times H \times W} \). 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 \( f : \mathbb{R}^{T \times 3 \times H \times W} \rightarrow \mathbb{R}^{(S \times T) \times 3 \times H \times W} \), which is built upon the exposure-aware representation \( u \) and the motion-aware representations \( m \) and \( n \) by progressive exposure-adaptive convolution and motion refinement. The architecture is shown in Fig. 2.

We first feed the input video sequence \( y \in \mathbb{R}^{(T \times 3) \times H \times W} \) into an encoder, which consists of four stages of residual blocks, separated by the spatially downsampling layers. Each residual block contains two \( 3 \times 3 \) 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,
### 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Deblurring</th>
<th>Interpolation</th>
<th>Avg</th>
<th>Deblurring</th>
<th>Interpolation</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDVD-TSP+QVI</td>
<td>33.15 / 0.951</td>
<td>31.28 / 0.926</td>
<td></td>
<td>35.63 / 0.970</td>
<td>36.12 / 0.973</td>
<td></td>
</tr>
<tr>
<td>CDVD-TSP+RIFE</td>
<td>30.97 / 0.925</td>
<td>30.33 / 0.913</td>
<td></td>
<td>33.36 / 0.952</td>
<td>33.06 / 0.952</td>
<td></td>
</tr>
<tr>
<td>MIMOUNetPlus+QVI</td>
<td>34.77 / 0.965</td>
<td>34.39 / 0.959</td>
<td></td>
<td>37.74 / 0.980</td>
<td>33.28 / 0.955</td>
<td></td>
</tr>
<tr>
<td>MIMOUNetPlus+RIFE</td>
<td>34.20 / 0.959</td>
<td>35.01 / 0.964</td>
<td></td>
<td>35.78 / 0.968</td>
<td>35.43 / 0.962</td>
<td></td>
</tr>
<tr>
<td>UTI-VFI</td>
<td>0.50 / 16.00</td>
<td>34.40 / 0.965</td>
<td></td>
<td>29.44 / 0.897</td>
<td>35.05 / 0.920</td>
<td></td>
</tr>
<tr>
<td>FLAVR</td>
<td>0.68 / 11.76</td>
<td>34.39 / 0.960</td>
<td></td>
<td>34.39 / 0.960</td>
<td>34.39 / 0.960</td>
<td></td>
</tr>
<tr>
<td>DeMFI</td>
<td>0.50 / 16.00</td>
<td>34.40 / 0.965</td>
<td></td>
<td>29.44 / 0.897</td>
<td>35.05 / 0.920</td>
<td></td>
</tr>
<tr>
<td>VIDUE (Ours)</td>
<td>0.277 / 29.63</td>
<td>36.32 / 0.974</td>
<td></td>
<td>36.12 / 0.973</td>
<td>35.56 / 0.970</td>
<td></td>
</tr>
</tbody>
</table>

The exposure-adaptive convolution is the input, whose channel factor and the bias term, followed by the exposure-adaptive convolution to obtain the feature representation. As in [23], we first perform instance normalization and the inter-motion maps $\hat{m}(l)$ is an element-wise multiplication operation.

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

$$g^{(l)} = \text{Refine}(\text{Warp}(g^{(l)}; \hat{n}(l))) + \text{Up}(g^{(l-1)}),$$

where $\text{Warp}(\cdot)$ denotes the backward warping function [35], and $\text{Up}(\cdot)$ denotes the bilinear upsampling operation. $\text{Refine}(\cdot)$ is implemented by a front-end convolution layer, two residual blocks, and a back-end convolution layer. To initialize progressive reconstruction, we set $g^{(0)} = 0$ as a tensor with all zeros, and summarize one stage of processing in Fig. 3. At last, we add $g^{(3)}$ (i.e., the output of the back-end residual block and convolution layer in the decoder) to $g^{(0)}$ to estimate the high framerate sharp video sequence $\hat{x} \in \mathbb{R}^{(5 \times T) \times 3 \times H \times W}$.

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.
### 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 ×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, $\Delta t_e : \Delta t \in \{5 : 8, 7 : 8\}$, 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 ×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 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 $g_o$ and $f$, and set the initial learning rate and the mini-batch size to $2 \times 10^{-4}$ and 12, respectively. We train the models for 200 epochs, and half 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 ×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.
### 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.

Additional, 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 ×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., MOUNetPlus+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 ×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.

---

<table>
<thead>
<tr>
<th>Method</th>
<th>GoPro-9:16</th>
<th>GoPro-11:16</th>
<th>GoPro-13:16</th>
<th>GoPro-15:16</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDVD-TSP+QVI</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>CDVD-TSP+RIFE</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>MIMOUNetPlus+QVI</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>MIMOUNetPlus+RIFE</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>FLAVR</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>DeMFI</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
<tr>
<td>VIDUE (Ours)</td>
<td>25.54 / 0.780</td>
<td>25.46 / 0.780</td>
<td>25.40 / 0.774</td>
<td>25.29 / 0.777</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Setting</th>
<th>GoPro-5:8</th>
<th>GoPro-7:8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exposure-Agnostic</td>
<td>34.79 / 0.965</td>
<td>34.13 / 0.960</td>
</tr>
<tr>
<td>Ordinal Regression</td>
<td>35.44 / 0.966</td>
<td>34.97 / 0.966</td>
</tr>
<tr>
<td>Contrastive Learning</td>
<td>36.32 / 0.974</td>
<td>35.63 / 0.970</td>
</tr>
<tr>
<td>Known Exposure Time</td>
<td>36.41 / 0.974</td>
<td>35.87 / 0.971</td>
</tr>
</tbody>
</table>

Table 4. Role of the exposure-aware feature representation evaluated by PSNR / SSIM. The default setting is highlighted in bold.
4.4. Evaluation on ×16 Interpolation Task

Without auxiliary information (e.g., event signals [25]), joint ×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 $u$: 1) one that is exposure-agnostic (i.e., without learning and adapting to $u$), 2) one that learns $u$ using ordinal regression, 3) one that learns $u$ 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 $\Delta t_e$ 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 ×8 and ×16 interpolation and deblurring tasks. Future work can be planned 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


[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