Bringing Events into Video Deblurring with Non-consecutively Blurry Frames

Wei Shang[1], Dongwei Ren[1]*, Dongqing Zou[2,5], Jimmy S. Ren[2,5], Ping Luo[3], Wangmeng Zuo[1,4]

[1]School of Computer Science and Technology, Harbin Institute of Technology
[2] SenseTime Research [3]The University of Hong Kong [4]Peng Cheng Laboratory, Shenzhen
[5]Qing Yuan Research Institute, Shanghai Jiao Tong University

Abstract

Recently, video deblurring has attracted considerable research attention, and several works suggest that events at high time rate can benefit deblurring. Existing video deblurring methods assume consecutively blurry frames, while neglecting the fact that sharp frames usually appear nearby blurry frame. In this paper, we develop a principled framework D^{2}Nets for video deblurring to exploit non-consecutively blurry frames, and propose a flexible event fusion module (EFM) to bridge the gap between event-driven and video deblurring. In D^{2}Nets, we propose to first detect nearest sharp frames (NSFs) using a bidirectional LSTM detector, and then perform deblurring guided by NSFs. Furthermore, the proposed EFM is flexible to be incorporated into D^{2}Nets, in which events can be leveraged to notably boost the deblurring performance. EFM can also be easily incorporated into existing deblurring networks, making event-driven deblurring task benefit from state-of-the-art deblurring methods. On synthetic and real-world blurry datasets, our methods achieve better results than competing methods, and EFM not only benefits D^{2}Nets but also significantly improves the competing deblurring networks.

1. Introduction

Videos have played the crucial role in computer vision field, and blur is commonly inevitable due to the movement of camera or moving objects in the capturing scene. To remedy the adverse effects of blur, video deblurring has drawn considerable research attention in many applications, *e.g.*, SLAM [12], 3D reconstruction [29] and tracking [36]. In recent years, event camera [3,22], a novel sensor for recording intensity changes of the capturing scene at microsecond level, has been developed, and events with high time rate are also suggested to facilitate deblurring [8, 25].

Both video deblurring [7, 9, 19] and event-driven deblurring [8, 20, 24, 25, 25, 33] have achieved unprecedented progresses, but they still have limitations. On the one hand, existing video deblurring networks are usually based on the assumption of consecutively blurry frames in a video, and design CNN-based [4, 11, 19, 30, 35, 38] and RNN-based [7, 18, 32] architectures, among which encoder-decoder architecture is the most popular choice to act as the the basic backbone. However, it is a common fact that blur does not consecutively occur in videos, *i.e.*, some frames in a blurry video are extremely sharp and clean [27]. These sharp frames actually can be exploited to facilitate the restoration of blurry frames, but they are indistinguishably processed in existing video deblurring methods, also adversely yielding sharp textures lost. On the other hand, event-driven restoration methods heavily rely on the employment of events, where various architectures such as BHA [20], CIE [28] and EDMD [8] are designed. In these methods, the modules for exploiting events are not easy to cooperate with existing image and video deblurring methods, thus restraining the development of principled framework for video deblurring

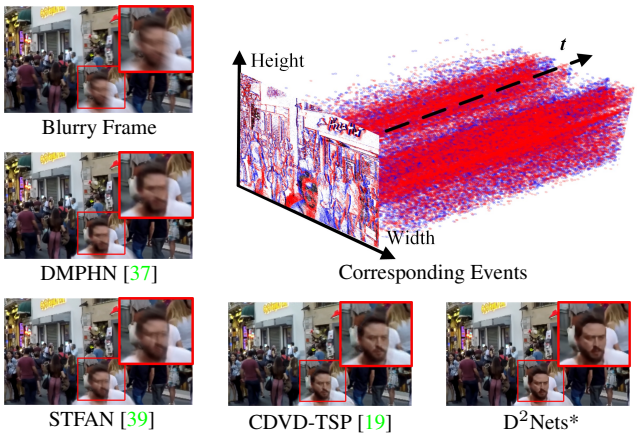

Figure 1. Deblurring results by our D^{2}Nets*, state-of-the-art image deblurring DMPHN [37] and video deblurring STFAN [39] and CDVD-TSP [19].

*Corresponding author: rendongweihit@gmail.com

and event-driven deblurring.

In this paper, we first develop a principled framework (*i.e.*, Detect&Deblur Netwotks, D^{2}Nets) to leverage non-consecutively blurry frames, and then propose an event fusion module (EFM) to bridge the gap between event-driven and video deblurring. First, our D^{2}Nets consists of three steps: (i) We propose to distinguish sharp frames and blurry frames using a bidirectional LSTM (BiLSTM) [6] as shown in Fig. 2, based on which two nearest sharp frames (NSFs) can be found for a blurry frame in the front and rear directions. BiLSTM can take as input either frames or their corresponding events. (ii) As shown in Fig. 3, blurry frames can then be restored to reconstruct latent sharp video frames using an encoder-decoder deblurring backbone, where NSFs are employed to guide the deblurring of a blurry frame instead of its neighboring frames. (iii) We further suggest to enhance the temporal consistency of restored video by a post-processing step, which is also beneficial to possibly surviving blurry frames due to detection errors by BiLSTM.

Second, the proposed EFM simultaneously exploits benefits from events and adjacent frames, and can be incorporated into the latent space of encoder-decoder architecture. EFM can then be incorporated into both steps of blurry frames restoration and temporal consistency enhancement in D^{2}Nets, bridging the gap between event-driven and video deblurring, resulting in D^{2}Nets*. Moreover, our EFM can be incorporated into existing state-of-the-art deblurring networks, *e.g.*, DMPHN [37], STFAN [39] and CDVD-TSP [19], making event-driven deblurring can benefit from these state-of-the-art image and video deblurring methods. As shown in Fig. 1, existing video deblurring methods cannot fully remove severe blur, while our D^{2}Nets* is able to restore more visually plausible deblurring result.

Experiments have been conducted on two benchmark datasets, including GoPro dataset [17] and Blur-DVS dataset [8] captured by DAVIS240C camera [3]. By exploiting NSFs, sharp textures from NSFs can better facilitate reconstructing latent clean frames, leading to notable gains by our D^{2}Net over state-of-the-art deblurring methods. The proposed EFM not only benefits our D^{2}Nets, but also significantly improves competing methods when collaborating with events to tackle video deblurring.

The contributions of this work are three-fold:

- A principled deblurring framework D^{2}Nets is developed to exploit nearest sharp frames when restoring blurry frames in a non-consecutively blurry video.
- An event fusion module EFM is proposed to better utilize beneficial information from events to facilitate deblurring.
- Our EFM has also been incorporated into existing image and video deblurring methods for tackling event-driven video deblurring. Extensive experiments are conducted to validate the effectiveness of D^{2}Nets and EFM for synthetic and real-world blurry videos.

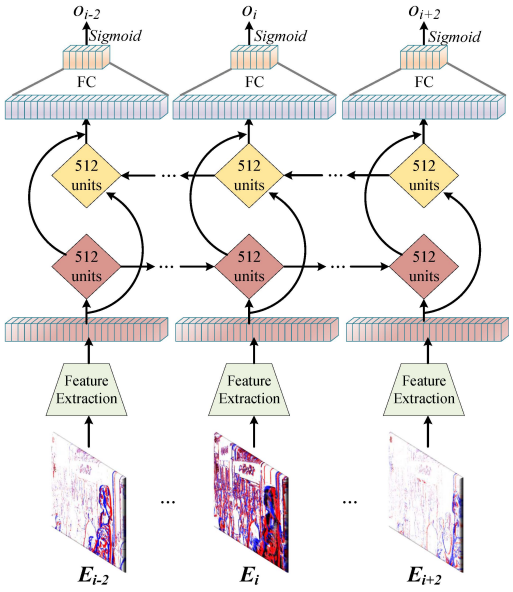

Figure 2. The architecture of BiLSTM detector for distinguishing sharp frames from blurry frames. The BiLSTM detector takes 5 adjacent frames as input, which can be either video frame sequence $\boldsymbol{B}_{i-2}, \cdots, \boldsymbol{B}_i, \cdots, \boldsymbol{B}_{i+2}$ or their corresponding events $\boldsymbol{E}_{i-2}, \cdots, \boldsymbol{E}_i, \cdots, \boldsymbol{E}_{i+2}$.

2. Related Work

In this section, we survey relevant works including image and video deblurring, and event-driven deblurring.

2.1. Image and Video Deblurring

Due to the success of encoder-decoder architecture in low-level vision field [14, 21], encoder-decoder is usually adopted as the most popular basic backbone in single image [4, 11, 32] deblurring and video deblurring [7, 18, 19, 30, 35, 39]. For single image deblurring, Tao *et al*. [32] proposed a scale-recurrent network in a "coarse-to-fine" scheme to extract multi-scale features from blurry image. Aittala *et al*. [1] designed an encoder-encoder architecture to treat all frames in the burst in an order-independent manner. Zhang *et al*. [37] presented a deep hierarchical multi-patch network inspired by spatial pyramid matching to deal with blurry images. Recently, Ren *et al*. [26] adopted an asymmetric autoencoder and a fully-connected network (FCN) to solve image deblurring in a self-supervised manner.

For video deblurring, Kim *et al*. [7] develop a spatial-temporal recurrent network with a dynamic temporal blending layer for latent frame restoration. To better leverage spatial and temporal information, Kim *et al*. [9] introduced an optical flow estimation step for aligning and aggregating

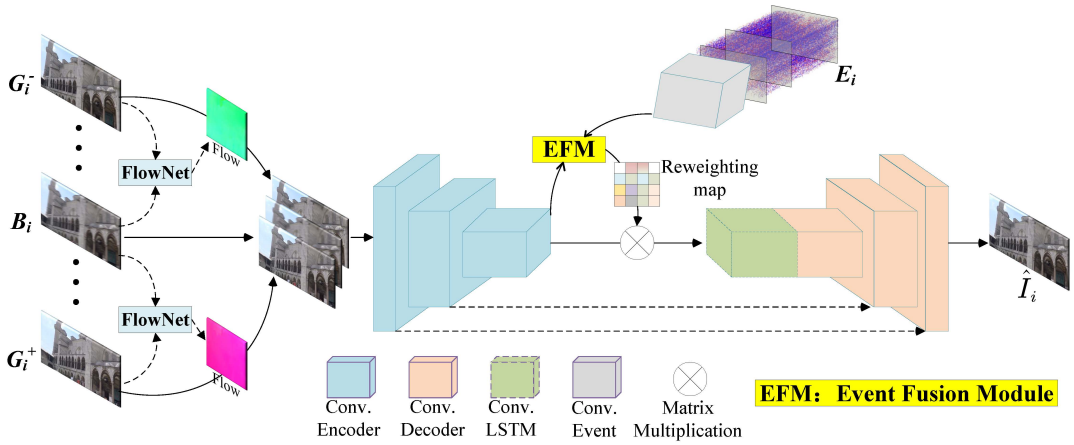

Figure 3. The flowchart of $\mathcal{F}_{\text{BRN}}$ for restoring blurry frame $\boldsymbol{B}_i$ with its NSFs $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$. $\mathcal{F}_{\text{BRN}}$ consists of two steps: warping NSFs aligned to $\boldsymbol{B}_i$ using f_{align}, and reconstructing latent frame $\hat{\boldsymbol{I}}_i$ using f_{rec}.

information across the neighboring frames to restore latent clean frame. In [34], Wang *et al*. developed deformable convolution in pyramid manner to implicitly align adjacent frames for better leveraging temporal information. Recently, Pan *et al*. [19] proposed to simultaneously estimate the optical flow and latent frames for video deblurring with the help of temporal sharpness prior. The estimated optical flow from intermediate latent frames as the motion blur information is fed back to the reconstruction network to generate final sharp frames.

Existing video deblurring methods assume consecutively blurry frames, which is commonly inconsistent with practical blurry videos. Ren *et al*. [27] found that some frames in a video with motion blur are sharp, and proposed to fit deblurring model to the test video. In this work, we propose a principled framework to better exploit sharp frames for video deblurring with non-consecutively blurry frames.

2.2. Event-driven Video Deblurring

Event cameras [3, 22] are novel sensors that record intensity changes of the scene at microsecond level with slight power consumption, and have potential applications in a variety of computer vision tasks, *e.g*., visual tracking [15], stereo vision [2] and optical flow estimation [13]. A related research branch is to explore the pure events to restore high frame rate image sequences [16, 25]. Recently, Pan *et al*. [20] formulated event-driven motion deblurring as a double integral model. Yet, the noisy hard sampling mechanism of event cameras often introduces strong accmulated noise and loss of scene details. Jiang *et al*. [8] proposed a sequential formulation of event-based motion deblurring, then unfolded its optimization steps as an end-to-end deep deblurring architecture. The employment of events is complicated, and existing methods leverage events in different ways as mentioned in [5]. Moreover, events modules in these methods are not trivial to incorporate into video deblurring networks, making it infeasible to benefit from state-of-the-art video deblurring methods. In this work, we propose an event fusion model, which can be easily incorporated into encoder-decoder architecture in existing image and video deblurring networks, bridging the gap between event-driven and video deblurring.

3. Proposed Method

In this section, we first present our principled Detect&Deblur (D^{2}Nets) framework for tackling video deblurring with non-consecutively blurry frames, and then elaborate the key components of D^{2}Nets in details.

3.1. Principled Framework of D^{2}Nets

For a blurry frame $\boldsymbol{B}_i$, the principled framework of D^{2}Nets can be formally presented as three steps,

$$\begin{aligned} \boldsymbol{G}_i^-, \boldsymbol{G}_i^+ &= \mathcal{F}_{\text{DET}}(\boldsymbol{B}_{i-N}, \cdots \boldsymbol{B}_i, \cdots \boldsymbol{B}_{i+N}), \\ \hat{\boldsymbol{I}}_i &= \mathcal{F}_{\text{BRN}}(\boldsymbol{G}_i^-, \boldsymbol{B}_i, \boldsymbol{G}_i^+), \\ \boldsymbol{I}_i &= \mathcal{F}_{\text{TCE}}(\hat{\boldsymbol{I}}_{i-1}, \hat{\boldsymbol{I}}_i, \hat{\boldsymbol{I}}_{i+1}). \end{aligned} \tag{1}$$

In D^{2}Nets: (i) $\mathcal{F}_{\text{DET}}$ detects nearest sharp frames (NSFs) $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$ from N adjacent frames in the front and rear orientations, respectively. (ii) $\mathcal{F}_{\text{BRN}}$ restores blurry frame guided by detected NSFs, which consists of an alignment module f_{align} for warping two NSFs $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$ aligned to $\boldsymbol{B}_i$ and an encoder-decoder f_{rec} to reconstruct latent sharp frame $\hat{\boldsymbol{I}}_i$ by fusing features of $\boldsymbol{B}_i$, $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$. (iii) $\mathcal{F}_{\text{TCE}}$ has the same two modules f_{align} and f_{rec}. $\mathcal{F}_{\text{TCE}}$ aims to further enhance the temporal consistency of frames $\{\hat{\boldsymbol{I}}_i\}_{i=1}^M$ and obtain the final deblurring video $\{\boldsymbol{I}_i\}_{i=1}^M$, where M is the total number of frames of the input video. The overall procedure of D^{2}Nets are presented in Alg. 1. The details of network architectures in $\mathcal{F}_{\text{DET}}$, $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$ can be found in supplementary file.

Moreover, we propose an event fusion module (EFM) to incorporate events into D^{2}Nets. In $\mathcal{F}_{\text{DET}}$, it is straightforward to substitute input frames as their corresponding events $\boldsymbol{E}_{i-N}, \cdots, \boldsymbol{E}_i, \cdots, \boldsymbol{E}_{i+N}$. In $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$, our EFM are flexible to incorporate into the encoder-decoder architectures. Besides, our EFM can also be incorporated into existing image and video deblurring networks, making them applicable to handle event-driven deblurring.

3.2. Detecting Blurry Frames and NSFs

BiLSTM Detector: We treat detecting blurry frames in a video as a binary classification task. Considering the temporal information in video, we in this paper propose to adopt bidirectional LSTM (BiLSTM) [6] to classify sharp frames and blurry frames, by which correlations of adjacent frames in both forward and backward directions are leveraged. The architecture of BiLSTM detector is visualized in Fig. 2. For a sequence of video frames, BiLSTM detector first extracts features using ResNet-152, and then transforms features to a 512-dimension vector as the input to BiLSTM. Finally, *Sigmoid* function is used to normalize the outputs of BiLSTM in the range [0,1], indicting a frame is blurry or sharp.

The consecutive frames in a blurry video are denoted by $\{\boldsymbol{B}_i\}_{i=1}^M$. Then, the output from detector is denoted by $\{o_i\}_{i=1}^M$, in which o_i is the probability of $\boldsymbol{B}_i$ being a sharp frame,

$$\{o_i\}_{i=1}^M = f_{\text{bilstm}}\left(\{\boldsymbol{B}_i\}_{i=1}^M\right), \tag{2}$$

where f_{bilstm} indicates the BiLSTM detector. To make the training easier, we split the video sequence into segments, each of which contains 5 frames. BiLSTM is trained by minimizing the binary cross-entropy loss function

$$\mathcal{L}_{\text{bilstm}} = -(o_i^{gt}\log(o_i) + (1-o_i^{gt})\log(1-o_i)), \tag{3}$$

where o_i^{gt} denotes the true label of i-th frame, *i.e.*, $o_i^{gt}=1$ when $\boldsymbol{B}_i$ is a sharp frame, otherwise $o_i^{gt}=0$.

Detecting NSFs: We binarize the outputs of BiLSTM by threshold $\epsilon = 0.5$. A frame $\boldsymbol{B}_i$ is blurry, if $o_i = 0$. Then for a given blurry frame $\boldsymbol{B}_i$, we can detect two NSFs $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$ from its N adjacent frames in the front and rear, respectively. If NSFs cannot be found, we simply set NSFs $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$ as the neighboring frames $\boldsymbol{B}_{i-1}$ and $\boldsymbol{B}_{i+1}$, respectively. In this work, we empirically set the searching range $N = 7$. This is because sharp frames beyond this range may have significant distinctions from the scene content in $\boldsymbol{B}_i$, and thus are not suitable to act as NSFs.

3.3. Blurry Frame Restoration with NSFs

After detecting two NSFs, blurry frame $\boldsymbol{B}_i$ can then be restored by $\mathcal{F}_{\text{BRN}}$. It is a natural strategy to direct take $(\boldsymbol{G}_i^-, \boldsymbol{B}_i, \boldsymbol{G}_i^+)$ as input in reconstruction network. However, there usually exists considerable temporal motion between $\boldsymbol{B}_i$ and its NSFs $\boldsymbol{G}_i^-$ and $\boldsymbol{G}_i^+$. Therefore, we implement $\mathcal{F}_{\text{BRN}}$ as two steps, as shown in Fig. 3, *i.e.*, warping NSFs aligned to blurry frame $\boldsymbol{B}_i$ using f_{align} and fusing their features to reconstruct latent frame using f_{rec}.

As for $f_{\text{align}}(\boldsymbol{G}_i^-, \boldsymbol{B}_i, \boldsymbol{G}_i^+)$, we use PWC-Net [31] as the optical flow estimation algorithm f_{flow} to provide motion composition,

$$\begin{aligned} \boldsymbol{u}_{\to i} &= f_{\text{flow}}\left(\boldsymbol{B}_i, \boldsymbol{G}_i^-\right),\ \boldsymbol{I}_i^- = \boldsymbol{G}^-\left(\boldsymbol{x} + \boldsymbol{u}_{\to i}\right) \\ \boldsymbol{u}_{i\leftarrow} &= f_{\text{flow}}\left(\boldsymbol{B}_i, \boldsymbol{G}^+\right),\ \boldsymbol{I}_i^+ = \boldsymbol{G}^+\left(\boldsymbol{x} + \boldsymbol{u}_{i\leftarrow}\right) \end{aligned} \tag{4}$$

where $\boldsymbol{u}_{\to i}$ and $\boldsymbol{u}_{i\leftarrow}$ are optical flow $\boldsymbol{G}_i^- \to \boldsymbol{B}_i$ and $\boldsymbol{B}_i \leftarrow \boldsymbol{G}_i^+$, respectively. The network f_{flow} is reused for two NSFs. Similar to [31], we use the bilinear interpolation to obtain the warped frames $\boldsymbol{I}_i^+$ and $\boldsymbol{I}_i^-$. Then blurry frames can be restored by

$$\hat{\boldsymbol{I}}_i = \begin{cases} f_{\text{rec}}\left(\boldsymbol{I}_i^+, \boldsymbol{B}_i, \boldsymbol{I}_i^-\right), & \text{if } o_i = 0 \\ \boldsymbol{B}_i, & \text{if } o_i = 1 \end{cases} \tag{5}$$

where f_{rec} is an encoder-decoder with LSTM to reconstruct the clean frame $\hat{\boldsymbol{I}}_i$.

As for training the parameters of f_{flow} and f_{rec}, we simultaneously update both of them, by minimizing the ℓ_1-norm loss function

$$\mathcal{L}_{\text{BRN}} = \sum_{i=1}^{K} \left\| \mathcal{F}_{\text{BRN}}\left(\boldsymbol{G}_i^-, \boldsymbol{B}_i, \boldsymbol{G}_i^+\right) - \boldsymbol{I}_i^{gt} \right\|_1, \tag{6}$$

where K is the number of detected blurry frames.

3.4. Temporal Consistency Enhancement

Using $\mathcal{F}_{\text{BRN}}$, blurry frames are usually restored without considering their neighboring frames, and the temporal consistency of whole video may be interfered. To remedy this problem, we further propose a temporal consistency enhancement network $\mathcal{F}_{\text{TCE}}$. In general, $\mathcal{F}_{\text{TCE}}$ shares the same two steps with $\mathcal{F}_{\text{BRN}}$, including frames alignment module and reconstruction module. The only distinction lies in the inputs of f_{align} and f_{rec},

$$\begin{aligned} (\boldsymbol{I}_i^-, \boldsymbol{I}_i^+) &= f_{\text{align}}(\hat{\boldsymbol{I}}_{i-1}, \hat{\boldsymbol{I}}_i, \hat{\boldsymbol{I}}_{i+1}), \\ \boldsymbol{I}_i &= f_{\text{rec}}\left(\boldsymbol{I}_i^-, \hat{\boldsymbol{I}}_i, \boldsymbol{I}_i^+\right), \end{aligned} \tag{7}$$

by which all the frames in latent video by $\mathcal{F}_{\text{BRN}}$ are enhanced by considering their neighboring frames. $\mathcal{F}_{\text{TCE}}$ not only can enhance the temporal consistency of restored video, but also will benefit possibly surviving blurry frames due to detection errors by BiLSTM, further improving the deblurring quality.

As for learning the parameters of $\mathcal{F}_{\text{TCE}}$, we also adopt ℓ_1-norm loss function

$$\mathcal{L}_{\text{TCE}} = \sum_{i=1}^{M} \left\| \mathcal{F}_{\text{TCE}}\left(\hat{\boldsymbol{I}}_{i-1}, \hat{\boldsymbol{I}}_i, \hat{\boldsymbol{I}}_{i+1}\right) - \boldsymbol{I}_i^{gt} \right\|_1. \tag{8}$$

Algorithm 1 D^{2}Nets (and D^{2}Nets*) for Video Deblurring

Input: Blurry video with M frames $\{\boldsymbol{B}_i\}_{i=1}^{M}$ (and optional events $\{\boldsymbol{E}_i\}_{i=1}^{M}$)
Output: Deblurring video $\{\boldsymbol{I}_i\}_{i=1}^{M}$

Initialize intermediate results $\{\hat{\boldsymbol{I}}_i\}_{i=1}^{M}$ as $\{\boldsymbol{B}_i\}_{i=1}^{M}$
// ***Lines 3-4:*** *$\mathcal{F}_{DET}$ detects blurry frames and NSFs*
Detect blurry frames $\{\boldsymbol{B}_j\}_{j=1}^{K}$ using BiLSTM.
Find NSFs $\boldsymbol{G}_j^-$ and $\boldsymbol{G}_j^+$ for $\boldsymbol{B}_j$, resulting in the set $\{\boldsymbol{G}_j^-, \boldsymbol{B}_j, \boldsymbol{G}_j^+\}_{j=1}^{K}$.
// ***Lines 6-9:*** *$\mathcal{F}_{BRN}$ restores detected blurry frames*
for $j = 1 : K$ **do**
    $(\boldsymbol{I}_j^-, \boldsymbol{I}_j^+) = f_{\text{align}}(\boldsymbol{G}_j^-, \boldsymbol{B}_j, \boldsymbol{G}_j^+)$
    $\hat{\boldsymbol{I}}_j = f_{\text{rec}}\left(\boldsymbol{I}_j^+, \boldsymbol{B}_j, \boldsymbol{I}_j^-\right)$
    Substitute corresponding frame in $\{\hat{\boldsymbol{I}}_i\}_{i=1}^{M}$ as $\hat{\boldsymbol{I}}_j$
end for
// ***Lines 11-13:*** *$\mathcal{F}_{TCE}$ enhances temporal consistency, and for the index exceeding range $[1, M]$, we simply repeat $\boldsymbol{B}_1$ or $\boldsymbol{B}_M$.*
for $i = 1 : M$ **do**
    $(\boldsymbol{I}_i^-, \boldsymbol{I}_i^+) = f_{\text{align}}(\hat{\boldsymbol{I}}_{i-1}, \hat{\boldsymbol{I}}_i, \hat{\boldsymbol{I}}_{i+1})$
    $\boldsymbol{I}_i = f_{\text{rec}}\left(\boldsymbol{I}_i^+, \hat{\boldsymbol{I}}_i, \boldsymbol{I}_i^-\right)$
end for
return Deblurring video frames $\{\boldsymbol{I}_i\}_{i=1}^{M}$

3.5. Event Fusion Module

As discussed in $\mathcal{F}_{\text{DET}}$, the events can be taken as input of BiLSTM detector to better distinguish sharp and blurry frames. We take one step further to leverage events in $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$, since events encode richer temporal information which is crucial for video deblurring task. For an event camera, given a blurry frame $\boldsymbol{B}_i$, its corresponding stream of events $\boldsymbol{E}_i$ are available. Each event has the form (t, x, y, p), which records intensity changes for coordinates (x, y) at time t, and polarity $p = \pm 1$ denotes the increase or decrease of intensity change. In this work, we transform the events stream into a tensor with 20 channels for each frame.

We propose EFM to better utilize rich boundaries in events for facilitating deblurring. Formally, our EFM can be presented as

$$\boldsymbol{m} = \mathit{SoftMax}\left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{W}_e^{\mathrm{T}}\boldsymbol{W}_z\boldsymbol{z}\right), \tag{9}$$

where $\boldsymbol{W}_e$ and $\boldsymbol{W}_z$ are learnable weight matrices, $\boldsymbol{z}$ is features of frames from latent space of encode-decoder, and $\boldsymbol{e}$ having same dimension with $\boldsymbol{z}$ is the feature of events $\boldsymbol{E}$ extracted using CNN along with downsampling. In EFM, $\boldsymbol{m}$ is a reweighting map which can be used to facilitate deblurring by matrix multiplication with the features of frames. EFM can be regarded as a kind of attention, where the reweighting map can guide the decoders mainly focus on specific features beneficial to deblurring. EFM is flexible to embed into the latent space of encoder-decoder like architecture in both $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$. Since PWCNet also adopts encoder-decoder like architecture, EFM can also be incorporated into f_{flow} to facilitate optical flow estimation. When cooperating with EFM, D^{2}Nets is denoted by D^{2}Nets*.

Moreover, considering that state-of-the-art deblurring methods, *e.g.*, DMPHN [37] for image deblurring, STFAN [39] and CDVD-TSP [19] for video deblurring, adopt encoder-decoder as their basic backbone, EFM can be easily incorporated into these methods, making event-driven deblurring benefit from state-of-the-art image and video deblurring methods.

4. Experiments

In this section, we evaluate our D^{2}Nets on two datasets, including GOPRO [17] and Blur-DVS [8]. D^{2}Nets is compared with state-of-the-art image deblurring method DMPHN [37], and video deblurring methods STFAN [39] and CDVD-TSP [19]. To evaluate their performance for event-driven deblurring, we apply our EFM into DMPHN, STFAN, CDVD-TSP and our D^{2}Nets, notated as DMPHN*, STFAN*, CDVD-TSP* and D^{2}Nets*, respectively. As for the methods specifically developed for event-driven deblurring, it is infeasible to fairly compare D^{2}Nets* with them quantitatively, since they usually do not release training codes. Therefore, we compare D^{2}Nets* with only one event-based method BHA [20] qualitatively, when handling real-world blurry frames. Our source code is available at https://github.com/shangwei5/D2Net.

4.1. Datasets and Training Details

4.1.1 Datasets

GoPro Dataset: First, we evaluate the competing methods on GoPro dataset [17], which is widely adopted for image deblurring and also is recently used in [20] to benchmark event-based deblurring. We follow [17, 20] to split the training and testing sets. To synthesize events, we use the open-source ESIM event simulator [23] to generate events based on sharp frames. To satisfy our assumption that sharp frames exist in a blurry video, we generate non-consecutively blurry frames in a video by randomly averaging adjacent sharp frames, *i.e.*, the average number is randomly chosen from 1 to 15. And we assume that a generated frame $\boldsymbol{B}_i$ is sharp if the number of averaging frames is smaller than 5, *i.e.*, $o_i^{gt} = 1$, otherwise $o_i^{gt} = 0$. It is worth noting that we randomly generate 50% blurry frames in a video, while the other 50% frames are sharp, without constraining that there must be 2 sharp ones in consecutive 7 frames.

Blur-DVS Dataset: To evaluate the competing methods when handling real-world events, we use Blur-DVS [8] captured by a DAVIS240C camera with a high speed event sensor and a low frame-rate Active Pixel Sensor for recording intensity frames at resolution 180×240. Blur-DVS includes

Table 1. Quantitative comparison of deblurring results of only blurry frames on GoPro dataset. * means that the method is incorporated with our EFM for exploiting events.

Method	DMPHN [37]	STFAN [39]	CDVD-TSP [19]	D^{2}Nets	DMPHN*	STFAN*	CDVD-TSP*	D^{2}Nets*
PSNR	26.70	26.01	26.29	**27.68**	26.86	27.19	**27.65**	27.39
SSIM	0.865	0.837	0.870	**0.906**	0.871	0.878	0.903	**0.907**

Table 2. Quantitative comparison of deblurring results of whole videos on GoPro dataset.

Method	DMPHN [37]	STFAN [39]	CDVD-TSP [19]	D^{2}Nets	DMPHN*	STFAN*	CDVD-TSP*	D^{2}Nets*
PSNR	31.58	30.12	30.31	**31.60**	31.90	30.90	**32.24**	31.76
SSIM	0.921	0.892	0.921	**0.940**	0.924	0.914	0.941	**0.943**

Table 3. The accuracy of BiLSTM detector by taking frames and events as input.

Input	Frames	Events
GoPro [17]	97.2%	99.0%
Blur-DVS [8]	94.8%	97.6%

Table 4. Component analysis on the GoPro dataset.

$\mathcal{F}_{\text{DET}}$	$\mathcal{F}_{\text{BRN}}$	$\mathcal{F}_{\text{TCE}}$	EFM	PSNR	SSIM
✗	✗	✓	✗	31.07	0.925
✓	✓	✗	✗	31.00	0.924
✓	✓	✓	✗	31.60	0.940
✓	✓	✓	✓	31.76	0.943

two subsets, *i.e.*, slow-motion subset and fast-motion subset. The slow-motion subset consists of 15,246 frames for relatively static scenes. When capturing, the camera movement is slow and stable, making blur rarely occurs in these collected frames. Thus, we can synthesize videos with non-consecutively blurry frames based on the slow-motion subset, similar with that on GOPRO dataset. We synthesize blurry videos by randomly averaging adjacent frames, *i.e.*, the averaging number varies from 1 to 9, based on which we can quantitatively compare these methods when handling real events. Finally, we obtain 2,029 pairs of blurry and sharp frames, among which 1,386 pairs are used for training, while 643 pairs are used for testing. Similarly, we assume that a generated frame $\boldsymbol{B}_i$ is sharp if the number of averaging frames is smaller than 5, *i.e.*, $o_i^{gt} = 1$, otherwise $o_i^{gt} = 0$. The fast-motion subset consists of 7 video sequences with 702 frames. When capturing, the camera movement is fast and unstable, and there are also moving objects. Thus, fast-motion subset can act as a real-world blurry testing set without ground-truth sharp frames.

4.1.2 Training Details

In the training process, we use ADAM optimizer [10] with parameters $\beta_1 = 0.9$, $\beta_1 = 0.999$, and $\epsilon = 10^{-8}$ for all the networks in $\mathcal{F}_{\text{DET}}$, $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$. The batch-size is set to be 12 and patch size is set to be 128×128. The training of networks in $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$ share the same hyper-parameters. In order to save training time, we use parameters of f_{rec} in $\mathcal{F}_{\text{BRN}}$ to initialize that in $\mathcal{F}_{\text{TCE}}$. For the PWC-Net f_{flow}, we adopt the pre-trained model [31] as initialization. The learning rates for reconstruction modules f_{rec} and optical flow estimation f_{flow} are initialized to be 1×10^{-4} and 1×10^{-6} respectively, and are decreased by multiplying 0.5 after every 100 epochs. The training ends after 250 epochs. For BiLSTM detector, the learning rate is set to be 1×10^{-4}, and the training ends after 100 epochs.

4.2. Ablation Study

4.2.1 Accuracy of BiLSTM Detector

Table 3 lists the accuracy of BiLSTM detector on two datasets by taking frames or events as input. One can see that BiLSTM detector with events as input are more precise than that with frames as input, since events naturally encode the motion trajectory. Nevertheless, the detection accuracy of these BiLSTM detectors is high, which is sufficient to find most blurry frames and their corresponding NSFs. We have also tried LSTM as the detector, and found notable accuracy decreases on GoPro, *i.e.*, 2.98% decrease for frames as input and 3.84% decrease for events as input. Also considering that BiLSTM (0.030*s* per frame) is not very inefficient in comparison to LSTM (0.028*s* per frame), BiLSTM is a better choice for the detector.

4.2.2 Effectiveness of Components

We evaluate the contribution of each component of D^{2}Nets on GOPRO dataset. As shown in Table 4 and Fig. 4, full D^{2}Nets achieves the best deblurring performance. We note that the individual $\mathcal{F}_{\text{TCE}}$ directly takes 3 neighboring frames as input. And it is interesting to find that individual $\mathcal{F}_{\text{TCE}}$ achieves higher PSNR than D^{2}Nets without $\mathcal{F}_{\text{TCE}}$. The reason can be attributed from two aspects: (i) Surviving blurry frames in $\mathcal{F}_{\text{DET}} + \mathcal{F}_{\text{BRN}}$ are not processed, and (ii) NSFs from long distance may have dramatic scene changes with the blurry frame, yielding temporal inconsistency. The results indicate that it is crucial to employ $\mathcal{F}_{\text{TCE}}$ to enhance temporal consistency, and it is necessary to include all the three components in D^{2}Nets. Furthermore, by incorporating EFM, deblurring performance can be further boosted. We also conducted experiment by replacing EFM as concatenation of events and frames, and obtained -0.42dB PSNR decrease on GoPro dataset. This is because events

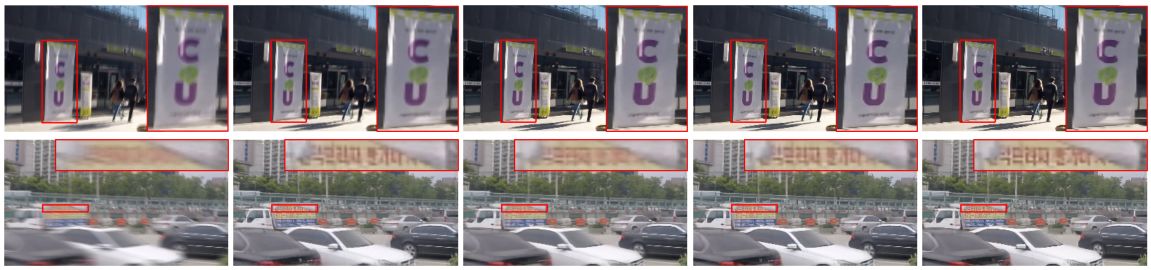

Figure 4. Visual comparison of component analysis on GoPro dataset. The first column is blurry frame, and 2 $\sim$ 5 columns correspond to the results of 1 $\sim$ 4 rows in Table 4. Zoom in for better view.

Figure 5. Visual comparison of deblurring results on GoPro dataset.

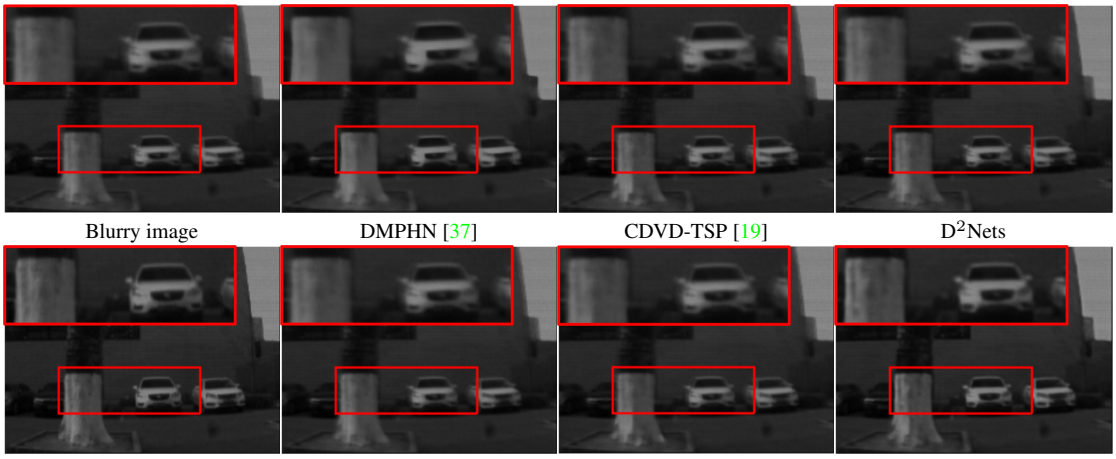

Figure 6. Visual comparison of deblurring results on slow-motion subset of Blur-DVS dataset.

contain rich spatial and temporal motion information, which cannot be fully exploited by naive concatenation.

4.3. Comparison with State-of-the-arts

We compare D^{2}Nets and D^{2}Nets* with state-of-the-art methods on GoPro and Blur-DVS datasets.

4.3.1 Evaluation on GoPro Dataset

On GoPro dataset, we evaluate the deblurring performance on both blurry frames (Table 1) and and whole video frames (Table 2). We retrain all these competing methods on our training dataset for a fair comparison. From the left side in Tables 1 and 2, one can see that our D^{2}Nets can achieve much higher quantitative metrics than the competing methods. This is because D^{2}Net can benefit from the nearest sharp frames, whose sharp texture details can be transferred to reconstruct latent clean frames. From the right side in Tables 1 and 2, D^{2}Nets* is still better than DMPHN*, STFAN* and CDVD-TSP* in terms of SSIM, which is more consistent with visual quality than PSNR. Moreover, our EFM can improve these competing video deblur-

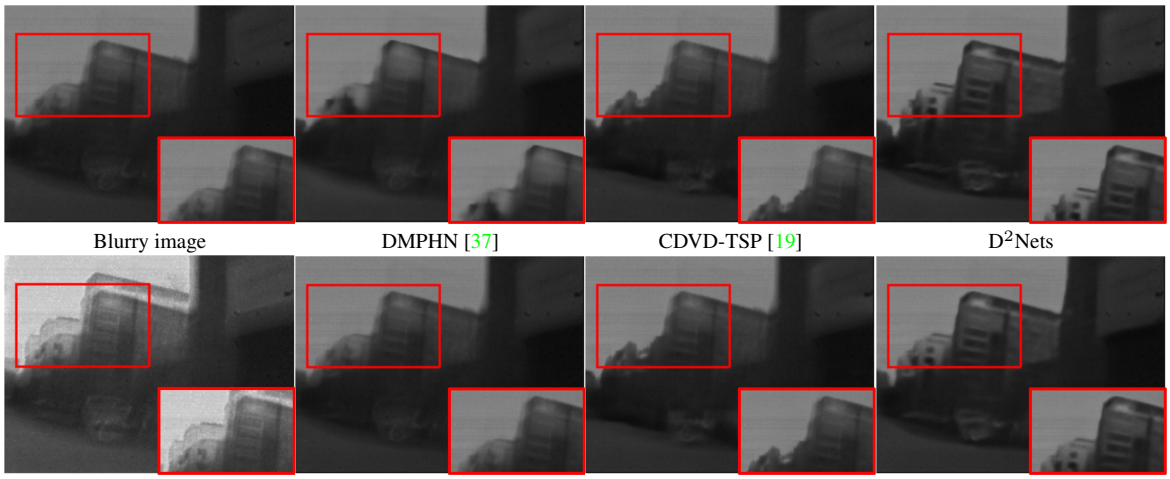

Figure 7. Visual comparison of deblurring results on fast-motion subset of Blur-DVS dataset. More results for real-world blurry frames can be found in supplementary file.

Table 5. Quantitative comparison of deblurring results of whole videos on Blur-DVS dataset. * means that the method is incorporated with our EFM for exploiting events.

Method	DMPHN [37]	STFAN [39]	CDVD-TSP [19]	D^{2}Nets	DMPHN*	STFAN*	CDVD-TSP*	D^{2}Nets*
PSNR	29.10	32.15	32.95	**33.96**	31.36	32.21	34.07	**34.24**
SSIM	0.808	0.827	0.811	**0.831**	0.812	0.827	0.811	**0.833**

ring methods to benefit from events. Especially for DMPHN* and CDVD-TSP*, their deblurring performances have been significantly boosted, validating the effectiveness of our EFM. In terms of visual quality comparison in Fig. 5, our D^{2}Nets can achieve sharper texture details, and plate license numbers are easily recognized than the results by the competing methods.

4.3.2 Evaluation on Blur-DVS Dataset

On Blur-DVS dataset, we only report quantitative results on whole video sequences, as shown in Table 5. We note that the competing methods and their versions with EFM are retrained on the training set of Blur-DVS. In Table 5, our D^{2}Nets and D^{2}Nets* achieve the best performance in comparison with their competing methods in terms of both PSNR and SSIM. Also DMPHN*, STFAN* and CDVD-TSP* obtain notable gains than their original versions when leveraging events using our EFM. Fig. 6 shows the visual quality comparison, from which one can see that our D^{2}Nets* can recover sharper texture details, due to the guidance of NSFs and events, while the results by other methods still suffer from mild blur or over-smoothing textures.

Finally, we evaluate these methods on real-world blurry frames from fast-motion subset of Blur-DVS. Besides DMPHN, STFAN and CDVD-TSP, we further take one event-based deblurring method BHA [20] into comparison. As shown in Fig. 7, our D^{2}Nets and D^{2}Nets* achieve the most visually plausible deblurring results with sharper textures, while DMPHN, STFAN and CDVD-TSP cannot fully remove severe blur. BHA recovers a blurry frame into a high-frame rate video guided by events. There are significant ringing artifacts around salient edge boundaries in deblurring result by BHA. In supplementary file, we provide more deblurring results of real-world blurry videos with events.

5. Conclusion

In this paper, we proposed a principled framework D^{2}Nets for tackling video deblurring with non-consecutively blurry frames. D^{2}Nets can better leverage possible sharp frames in blurry videos, benefiting from which blurry frames can be better restored and the temporal consistency of deblurring video can be encouraged. We further proposed a flexible event fusion module (EFM), which can be incorporated into not only our D^{2}Nets but also existing image and video deblurring networks. Our EFM makes event-driven deblurring task benefit from state-of-the-art image and video deblurring networks, and will be extended to event-driven super-resolution and interpolation tasks in future work.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61801326 and U19A2073. This work was also supported by SenseTime Research Fund for Young Scholars.

References

[1] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *European Conference on Computer Vision*, pages 731–747, 2018. 2

[2] Alexander Andreopoulos, Hirak J Kashyap, Tapan K Nayak, Arnon Amir, and Myron D Flickner. A low power, high throughput, fully event-based stereo system. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7532–7542, 2018. 3

[3] 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 Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. 1, 2, 3

[4] Huaijin Chen, Jinwei Gu, Orazio Gallo, Ming-Yu Liu, Ashok Veeraraghavan, and Jan Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *IEEE International Conference on Computational Photography*, pages 1–9, 2018. 1, 2

[5] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *IEEE International Conference on Computer Vision*, pages 5633–5643, 2019. 3

[6] Sepp Hochreiter and Michael C Mozer. A discrete probabilistic memory model for discovering dependencies in time. In *International Conference on Artificial Neural Networks*, pages 661–668, 2001. 2, 4

[7] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *IEEE International Conference on Computer Vision*, pages 4038–4047, 2017. 1, 2

[8] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. 1, 2, 3, 5, 6

[9] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *European Conference on Computer Vision*, pages 106–122, 2018. 1, 2

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 6

[11] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2018. 1, 2

[12] Hee Seok Lee, Junghyun Kwon, and Kyoung Mu Lee. Simultaneous localization, mapping and deblurring. In *IEEE International Conference on Computer Vision*, pages 1203–1210, 2011. 1

[13] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. *British Machine Vision Conference*, 2018. 3

[14] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. pages 2802–2810, 2016. 2

[15] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1–9, 2018. 3

[16] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018. 3

[17] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 2, 5, 6

[18] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. Recurrent neural networks with intra-frame iterations for video deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8102–8111, 2019. 1, 2

[19] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3043–3051, 2020. 1, 2, 3, 5, 6, 7, 8

[20] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019. 1, 3, 5, 8

[21] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[22] Lichtsteiner Patrick, Christoph Posch, and Tobi Delbruck. A 128x 128 120 db 15μ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-state Circuits*, 43:566–576, 2008. 1, 3

[23] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, pages 969–982, 2018. 5

[24] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019. 1

[25] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 3

[26] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020. 2

[27] Xuanchi Ren, Zian Qian, and Qifeng Chen. Video deblurring by fitting to test data. In *European Conference on Computer Vision*, 2020. 1, 3

[28] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*, pages 308–324, 2018. 1

[29] Hee Seok Lee and Kuoung Mu Lee. Dense 3d reconstruction from severely blurred images using a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1

[30] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 1, 2

[31] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 4, 6

[32] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 1, 2

[33] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, 2019. 1

[34] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[35] Patrick Wieschollek, Michael Hirsch, Bernhard Scholkopf, and Hendrik Lensch. Learning blind motion deblurring. In *IEEE International Conference on Computer Vision*, pages 231–240, 2017. 1, 2

[36] Yi Wu, Haibin Ling, Jingyi Yu, Feng Li, Xue Mei, and Erkang Cheng. Blurred target tracking by blur-driven tracker. In *IEEE International Conference on Computer Vision*, 2011. 1

[37] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 1, 2, 5, 6, 7, 8

[38] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *IEEE Transactions on Image Processing*, 28(1):291–301, 2018. 1

[39] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *IEEE International Conference on Computer Vision*, pages 2482–2491, 2019. 1, 2, 5, 6, 7, 8