Bringing Events into Video Deblurring with Non-consecutively Blurry Frames
Supplementary Material

In this supplementary file, we provide the details of network architecture, more results of ablation study, more results on benchmark datasets and real-world blurry videos.

1. Network Architecture

1.1. Architectures of D^{2}Nets

D^{2}Nets includes BiLSTM Detector in $\mathcal{F}_{\text{DET}}$ to detect blurry frames in a video, $\mathcal{F}_{\text{BRN}}$ to restore blurry frame guided by detected NSFs and $\mathcal{F}_{\text{TCE}}$ to enhance the temporal consistency of restored video. EFM is proposed to better utilize rich boundaries in events for facilitating deblurring, which can be embeded in $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$. We first formulate EFM, and then detail the network architecture.

1.1.1 Event Fusion Module (EFM)

To reduce the computational cost of EFM, we suggest to incorporate EFM into the bottleneck of encoder-decoder, which can be formulated as

$$\begin{aligned} \boldsymbol{z} &= \mathcal{F}_{\text{Encoder}}(\boldsymbol{I}_i^+, \boldsymbol{B}_i, \boldsymbol{I}_i^-), \\ \boldsymbol{e} &= \mathcal{F}_{\text{Event}}(\boldsymbol{E}_i), \\ \boldsymbol{m} &= \mathit{SoftMax}\left(\left(\mathcal{R}\left(\boldsymbol{W}_e\boldsymbol{e}\right)\right)^{\text{T}} \otimes \mathcal{R}\left(\boldsymbol{W}_z\boldsymbol{z}\right)\right), \\ \boldsymbol{y} &= \boldsymbol{z} + \mathcal{R}'\left(\mathcal{R}\left(\boldsymbol{W}_v\boldsymbol{z}\right) \otimes \boldsymbol{m}\right), \end{aligned} \tag{1}$$

where $\otimes$ is matrix multiplication, $\mathcal{F}_{\text{Encoder}}$ is convolutional Encoder, $\mathcal{F}_{\text{Event}}$ is convolutional module that can guarantee $\boldsymbol{e}$ has the same spatial dimension with $\boldsymbol{z}$, $\mathcal{R}$ is the reshape operation that changes dimension from (c, h, w) to $(c, h * w)$, and $\mathcal{R}'$ is the reshape operation that changes dimension from $(c, h*w)$ to (c, h, w). $\boldsymbol{z}$ is features of frames from latent space of encoder, and $\boldsymbol{e}$ is the feature of events $\boldsymbol{E}_i$ extracted using CNN along with downsampling. $\boldsymbol{m}$ is a reweighting map which can be used to facilitate deblurring by matrix multiplication with the features of frames. $\boldsymbol{W}_e$, $\boldsymbol{W}_z$ and $\boldsymbol{W}_v$ are learnable weight matrices. Finally, $\boldsymbol{y}$ is the input of decoder to recover latent sharp frame.

1.1.2 Architecture details

D^{2}Nets includes three parts: $\mathcal{F}_{\text{DET}}$, $\mathcal{F}_{\text{BRN}}$ and $\mathcal{F}_{\text{TCE}}$. Since $\mathcal{F}_{\text{TCE}}$ shares the same architecture with $\mathcal{F}_{\text{BRN}}$, we only present architecture details of $\mathcal{F}_{\text{BRN}}$. BiLSTM detector in $\mathcal{F}_{\text{DET}}$ is detailed in Table s1, $\mathcal{F}_{\text{BRN}}$ is detailed in Table s3. $\mathcal{F}_{\text{Event}}$ is detailed in Table s4, which can be embeded into the bottleneck of $\mathcal{F}_{\text{BRN}}$ or $\mathcal{F}_{\text{TCE}}$.

Table s1. The architecture of BiLSTM detector in $\mathcal{F}_{\text{DET}}$. Convolution is with the form Conv(input channel, output channel, kernel size, stride, padding size) and fully connected layer is with the form Linear(input channel, output channel).

Input	A sequence of frames (indim = 3) or events (indim = 20);
Layer f_{in}	Conv(indim,3, 3, 1, 1); LeakyReLU;
Layer $f_{extractor}$	*ResNet-152*;
Layer fc_1	Linear(2048, 512);
Layer f_{rec}	BiLSTM;
Layer fc_2	Linear(1024, 1); Sigmoid;
Output	A sequence of probabilities of inputs being sharp frames;

Table s2. The basic component ResBlock. Convolution is with the form Conv(input channel, output channel, kernel size, stride, padding size). *feats* is the number of channels from previous layer.

Input	Feature maps from the previous layer;
Layer 1	Conv(*feats*, *feats*, 3, 1, 1); LeakyReLU;
Layer 2	Conv(*feats*, *feats*, 3, 1, 1);
	Add(*Input*, *Layer2*);

Table s3. The architecture of f_{rec} in $\mathcal{F}_{\text{BRN}}$, while we omit the arthtecture of f_{flow} due to its exactly same architecture with PWC-Net [5]. Convolution is with the form Conv(input channel, output channel, kernel size, stride, padding size) and deconvolution is with the form ConvTrans(input channel, output channel, kernel size, stride, padding size).

Input: Front NFS $\boldsymbol{G}_i^-$, blurry frame $\boldsymbol{B}_i$, near NFS $\boldsymbol{G}_i^+$	
Frames aligning:	
Warping NSFs aligned to blurry frame $\boldsymbol{B}_i$ using f_{flow}.	
Output: the warped frames $\boldsymbol{I}_i^-$ and $\boldsymbol{I}_i^+$	
Concatenate($\boldsymbol{I}_i^-$, $\boldsymbol{B}_i$, $\boldsymbol{I}_i^+$)	
Encoder:	
Layer f_{in}	Conv(9, 32, 3, 1, 1); LeakyReLU; 3 ResBlocks;
Layer f_{en1}	Conv(32, 64, 3, 2, 1); LeakyReLU; 3 ResBlocks;
Layer f_{en2}	Conv(64, 128, 3, 2, 1); LeakyReLU; 3 ResBlocks;
Bottleneck:	
Layer f_{rec}	*LSTM*;
Decoder:	
Layer f_{de2}	3 ResBlocks; ConvTrans(128, 64, 3, 2, 1); LeakyReLU;
Layer f_{de1}	3 ResBlocks; ConvTrans(64, 32, 3, 2, 1); LeakyReLU;
Layer f_{out}	3 ResBlocks; Conv(32, 3, 3, 1, 1);
Output: The latent sharp frame $\hat{\boldsymbol{I}}_i$	

Table s4. The architecture of $\mathcal{F}_{\text{Event}}$. Convolution is with the form Con.(input channel, output channel, kernel size, stride, padding size). UnPS(factor) is a reverse operation of pixel-shuffle to perform downsampling, and we use factor $= 2$ to guarantee that the feature size is same with $\boldsymbol{z}$.

Input	Events $\boldsymbol{E}_i$;
Layer f_{event_in1}	Conv(20,32,1,1,0); LeakyReLU;
Layer f_{event_in2}	Conv(32,32,3,2,1); LeakyReLU;
Layer f_{ups}	UnPS(2);
Output	Events feature $\boldsymbol{e}$;

2. Ablation Study

2.1. Effects of Components in D^{2}Nets

Here, we present the deblurring results by D^{2}Nets and its variants. One can see that full D^{2}Nets* obtain visually favorable results as shown in Fig. s1.

(a) Blurry Franmes

(b) Results by $\mathcal{F}_{\text{TCE}}$

(c) Results by $\mathcal{F}_{\text{DET}}$+$\mathcal{F}_{\text{BRN}}$

(d) Results by $\mathcal{F}_{\text{DET}}$+$\mathcal{F}_{\text{BRN}}$ + $\mathcal{F}_{\text{TCE}}$

(e) Results by Full D^{2}Nets*

Figure s1. Visual comparison of component analysis on GoPro dataset. The first row is blurry frames, and (b)∼(e) correspond to the results of 1 ∼ 4 rows of Table 4 in the manuscript.

2.2. Effectiveness of Event Fusion Module

In Fig. s2, we visualize the feature map in bottleneck with and without our EFM, respectively.

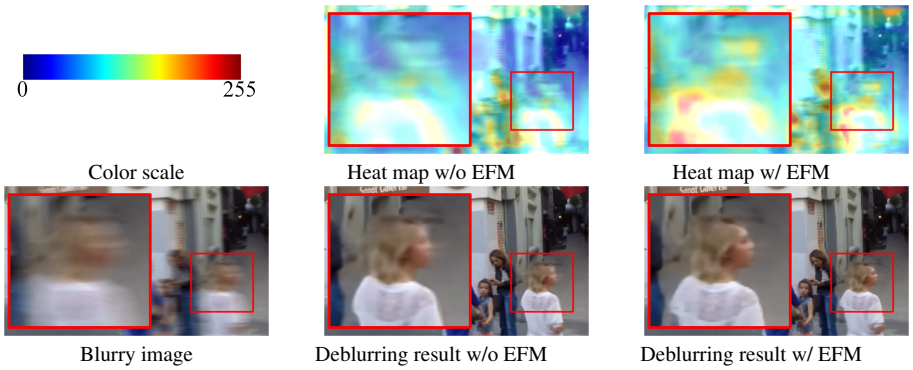

Figure s2. Effect of EFM. The feature map reweighted by our EFM pays more attention on blurry boundaries, which benefits the sharp frame restoration.

3. More Results on Benchmark Datasets

3.1. More Results on GoPro Dataset [2]

The results by DMPHN [6] and CDVD-TSP [3] cannot fully remove severe blur, while our results are more visually plausible.

Blurry frames

Results by DMPHN

Results by CDVD-TSP

Results by our D^{2}Nets*

Figure s3. More results on GoPro dataset. From 1 to 4 rows are blurry frames, DMPHN [6], CDVD-TSP [3] and D^{2}Net*, respectively.

3.2. More Results on Blur-DVS Dataset [1]

The results by DMPHN [6] and CDVD-TSP [3] still have serve blur.

Figure s4. More results comparison on Blur-DVS dataset. From 1 to 4 rows are Blurry frames, DMPHN [6], CDVD-TSP [3] and D^{2}Net*, respectively.

3.3. More Results on Severe Blurry Frames

We present more results on severe blurry frames with events from fast-motion subset of Blur-DVS dataset.

The left is blurry frames, the middle is the results by BHA [4], and the right is the results by our D^{2}Nets*

References

[1] 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. 5

[2] 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. 4

[3] 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. 4, 5

[4] 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. 6

[5] 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. 2

[6] 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. 4, 5