

# Learning Event-Based Motion Deblurring

Zhe Jiang<sup>\*1,2</sup> Yu Zhang<sup>\*†1,3</sup> Dongqing Zou<sup>1</sup> Jimmy Ren<sup>1</sup> Jiancheng Lv<sup>2</sup> Yebin Liu<sup>3</sup>  
<sup>1</sup>SenseTime Research <sup>2</sup>Sichuan University <sup>3</sup>Department of Automation, Tsinghua University

## Abstract

Recovering sharp video sequence from a motion-blurred image is highly ill-posed due to the significant loss of motion information in the blurring process. For event-based cameras, however, fast motion can be captured as events at high time rate, raising new opportunities to exploring effective solutions. In this paper, we start from a sequential formulation of event-based motion deblurring, then show how its optimization can be unfolded with a novel end-to-end deep architecture. The proposed architecture is a convolutional recurrent neural network that integrates visual and temporal knowledge of both global and local scales in principled manner. To further improve the reconstruction, we propose a differentiable directional event filtering module to effectively extract rich boundary prior from the stream of events. We conduct extensive experiments on the synthetic GoPro dataset and a large newly introduced dataset captured by a DAVIS240C camera. The proposed approach achieves state-of-the-art reconstruction quality, and generalizes better to handling real-world motion blur.

## 1. Introduction

Motion blur happens commonly due to the exposure time required by modern camera sensors, during which scenes are recorded at different time stamps and accumulated into averaged (blurred) signal. The inverse problem called *deblurring*, which unravels the underlying scene dynamics behind a motion-blurred image and generates a sequence of sharp recovery of the scene, is still challenging in computer vision. While simple motion patterns (*e.g.* camera shake) have been well modelled [29, 22, 6, 8, 47, 12, 51, 2], formulating more sophisticated motion patterns in real world, however, is much more difficult.

To model general motion blur, recent deep learning approaches propose to recover a blurred image by observing lots of sharp images and their blurred versions [42, 11, 52, 25, 15, 43]. Despite their success in certain scenarios, they

<sup>\*</sup>Equal contribution. This work is done when Jiang Zhe is interned with SenseTime Research.

<sup>†</sup>Correspondence should be addressed to: zhangyulb@gmail.com

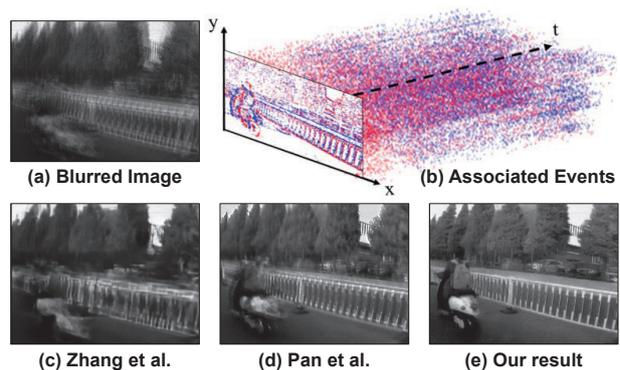


Figure 1. Motivation of our approach. A severe motion-blurred image (a) is difficult to deblur by observing its ambiguous appearance only even with state-of-the-art deep architecture [50] (c). Though events (b) provide dense temporal cues, the physical reconstruction approach [31] still presents unaddressed blur due to the noisiness of events (d). The proposed deep motion deblurring learns to recover plausible details from imperfect image and events (e).

may fail reconstructing the scene plausibly for severe motion blur (*e.g.* Fig. 1), which is common for handheld, vehicle or drone-equipped cameras. In this case, hallucinating the scene details is hardly possible due to the significant loss of temporal order and visual information.

Instead of purely relying on computational architectures, this work adopts event-based cameras to alleviate this problem at data capture stage. Event cameras are biologically inspired sensors adept at recording the change of pixel intensities (called *events*) with microsecond accuracy and very low power consumption. The hybrid model of such sensors (*e.g.* [5]) allows the events being temporally calibrated with the image. As a result, such data naturally encodes dense temporal information that can facilitate motion deblurring. As shown in Fig. 1 (a) and (b), although the image undergoes significant blur, the accompanying events are temporally dense and reveal clear moving pattern of the scene.

Despite the high potential of event-based motion deblurring, a critical issue is that events are lossy and noisy signals triggered only if pixel intensity changes up to certain threshold that can vary with the change of scene conditions [35]. Such discrete and inconsistent sampling makes textures and

contrast difficult to restore. As shown in Fig. 1 (d), state-of-the-art physical deblurring method [31] still has difficulty reconstructing the image plausibly. Our solution is to plug deeply learned priors into event-based deblurring process, so as to surpass the imperfectness of data.

In details, this work starts from a sequential formulation of event-based deblurring. By reinterpreting its optimization with deep networks, we propose a novel recurrent architecture trainable end-to-end. For each time step, coarse reconstructions are obtained from previous reconstruction as well as the local temporal events. Fine details are then supplied by network predictions, guided by appearance and temporal cues at both global and local scales. To further improve the quality of reconstruction, we propose a differentiable Directional Event Filtering (DEF) module, which effectively aggregates the motion boundaries revealed by events and produces sharp deblurring prior. To evaluate the proposed approach, we compile a large outdoor dataset captured using the DAVIS240C camera [5]. Extensive experiments on this dataset and the synthetic GoPro dataset [25] show that the proposed approach outperforms various state-of-the-art methods, either image-based or event-based, and generalizes better to handling real-world motion blur.

Contributions of this paper are summarized as follows.

- 1) We propose a novel recurrent deep architecture for event-based motion deblurring, which achieves state-of-the-art results on two large benchmarks.
- 2) We propose directional event filtering to generate sharp boundary prior from events for motion deblurring.
- 3) We compile a new event dataset with real-world motion blur to facilitate future research.

## 2. Related Work

**Blind motion deblurring** aims to resolve a blurry image without knowing the blurring kernel. Early works have designed various blurring-aware indicators, such as color channel statistics [29, 47], patch recurrence [22] and “outlier” image signals [8], to define latent image priors. Several works propose to learn motion kernels [39, 28], restoration functions [45, 11] and image priors [55, 42] from data. More complex motion patterns compounded by different objects were also addressed [16, 37]. Richer prior knowledge such as scene geometry was proven useful [30, 32].

A recent trend is to approach all the complexities of motion deblurring with deep neural networks. Various kinds of effective network designs are proposed, including enlarging the receptive field [52], multi-scale fusion [25, 27], feature distangling [26], and recurrent refinement [44]. There was also research on decoding the motion dynamics of a blurred image to a sharp video sequence [15]. Despite these advances, the considerable combinations of real-world lightings, textures and motions, which are severely missing in a blurred image, are still difficult to be plausibly recovered.

**Event cameras** [19, 5] are a special kind of sensors that

detect intensity changes of the scene at microsecond level with slight power consumption. They find applications in various vision tasks, such as visual tracking [34, 23], stereo vision [54, 1] and optical flow estimation [20, 48]. A related branch is to explore the corrupted event signals to restore high frame rate image sequences [38, 24, 40]. Recently, Pan *et al.* [31] formulates event-based motion blurring with a double integral model. Yet, the noisy hard sampling mechanism of event cameras often introduces strong accumulated noise and loss of scene details/contrast.

This work shares the insight of recent works on event-to-video translation [33, 17, 36] that surpasses the imperfect event sampling by learning plausible details from data. While [33] addresses future frame prediction, [17, 36] translate events to plausible intensity images in streaming manner depending on local motion cues. Instead, this work explores both long-term, local appearance/motion cues as well as novel event boundary priors to solve motion deblurring.

## 3. Learning Event-Based Motion Deblurring

Given a motion-blurred image  $\bar{\mathcal{I}}$ , our objective is to recover a sharp video sequence with  $T$  frames,  $\mathbb{I} = \{\mathcal{I}_i\}_{i=1}^T$ . We assume that a set of events  $\mathbb{E}_{1 \sim T}$  are also captured by hybrid image-event sensors during the exposure, where the tilde denotes the time interval. Each event  $\mathcal{E} \in \mathbb{E}_{1 \sim T}$  has the form  $\mathcal{E}_{x,y,t}$ , meaning that it is triggered at image coordinate  $(x, y)$  and time point  $t \in [1, T]$ . Note here  $t$  does not need to be an integer, but can be fractional due to the high temporal resolution (*i.e.* microsecond-level) of event camera. A polarity  $p_{x,y,t}$  is recorded for  $\mathcal{E}_{x,y,t}$  indicating the change of local intensity. Formally, it is defined as [19, 5]

$$p_{x,y,t} = \begin{cases} +1, & \text{if } \log \left( \frac{\mathcal{I}_t(x,y)}{\mathcal{I}_{t-\Delta t}(x,y)} \right) > \tau, \\ -1, & \text{if } \log \left( \frac{\mathcal{I}_t(x,y)}{\mathcal{I}_{t-\Delta t}(x,y)} \right) < -\tau, \end{cases} \quad (1)$$

Eqn. (1) shows that, events are triggered if the instant image at time point  $t$ , namely  $\mathcal{I}_t$ , has pixel intensity changed up to a threshold  $\pm\tau$  in a small time period  $\Delta t$ . Without loss of generality, we assume that  $p_{x,y,t}$  takes zero in case that  $\log \left( \frac{\mathcal{I}_t(x,y)}{\mathcal{I}_{t-\Delta t}(x,y)} \right)$  is in  $[-\tau, \tau]$ . For adjacent latent images  $\mathcal{I}_i$  and  $\mathcal{I}_{i-1}$ , the following relationship can be derived:

$$\mathcal{I}_i(x, y) \approx \mathcal{I}_{i-1}(x, y) \cdot \exp \left( \tau \int_{t=i-1}^i p_{x,y,t} \mathbb{1}(\mathcal{E}_{x,y,t}) dt \right), \quad (2)$$

The indicator function  $\mathbb{1}(\cdot)$  equals 1 if the event  $\mathcal{E}_{x,y,t}$  exists, or 0 otherwise.

One should note that the approximation error of (2) is getting lower when  $\Delta t, \tau \rightarrow 0$ , which implies denser events according to (1). However, with inconsistent  $\tau$  affected by various kinds of noise, the approximation is mostly insufficient in practice, leading to loss of contrast and details. To

address this issue, we propose a joint framework that learns to reconstruct clean images from data, by reinterpreting a sequential deblurring process.

**Deep sequential deblurring.** Event-assisted deblurring can be formulated under Maximum-a-Posteriori:

$$\mathbb{I}^* = \arg \max_{\mathbb{I}} P(\mathbb{I} | \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T}). \quad (3)$$

To solve the combinatorial problem (3) we make the following simplifications. For the joint posterior  $P(\mathbb{I} | \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T})$ , we make use of the temporal relations between adjacent latent images (2), and assume a Markov chain model:

$$P(\mathbb{I} | \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T}) \approx P(\mathcal{I}_T | \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T}) \times \prod_{i=1}^{T-1} P(\mathcal{I}_i | \mathcal{I}_{i+1}, \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T}), \quad (4)$$

in which  $P(\mathcal{I}_i | \mathcal{I}_{i+1}, \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T}) = P(\mathcal{I}_i | \mathcal{I}_{i+1}, \bar{\mathcal{I}}, \mathbb{E}_{i \sim i+1})$  with Markov assumption. Note that this simplified model first estimates  $\mathcal{I}_T$ , then perform sequential reconstruction in backward order. According to Bayesian rule, the maximizer of a backward reconstruction step equals to:

$$\mathcal{I}_i^* = \arg \max_{\mathcal{I}_i} P(\mathcal{I}_{i+1}, \bar{\mathcal{I}}, \mathbb{E}_{i \sim i+1} | \mathcal{I}_i) P(\mathcal{I}_i). \quad (5)$$

Here, the prior term  $P(\mathcal{I}_i)$  imposes desired distributions of the latent image, e.g.  $\ell_1$  gradient [3] or manifold smoothness [24] in recent event-based image reconstruction. To model the likelihood term, we assume that there is an initial estimate from previous reconstruction, via (2):

$$\hat{\mathcal{I}}_i = \mathcal{I}_{i+1} \odot \exp(-\tau \mathcal{S}_{i+1}^i), \quad (6)$$

where  $\forall x, y, \mathcal{S}_{i+1}^i(x, y) = \int_{t=i}^{i+1} p_{x,y,t} \mathbb{1}(\mathcal{E}_{x,y,t}) dt$ , and  $\odot$  denotes Hadamard product. Since the time interval is small, we assume constant  $\tau$  which introduces only small drift and provides good initialization. To solve  $\mathcal{I}_i^*$ , several works assume simple distributions centered around  $\hat{\mathcal{I}}_i^*$  to define the likelihood term in (5), e.g. in [24] a Poisson distribution is used. In this manner, Eqn. (5) can be treated as a well-studied denoising problem.

Instead of using simple image prior, we borrow from recent research on learning deep denoising prior [53, 50]. In particular, we plug a deep network  $\mathcal{N}$  as a learned denoiser,

$$\mathcal{I}_i^* = \mathcal{N}(\hat{\mathcal{I}}_i, \mathcal{I}_{i+1}, \bar{\mathcal{I}}, \mathbb{E}_{i \sim i+1}). \quad (7)$$

As such, prior of latent image  $P(\mathcal{I}_i)$  is not explicitly defined but implicitly learned from training data. To reduce parameter size and prevent overfitting, we use the same network governed by the same set of parameters for each deblurring step of (5), leading to a recurrent architecture.

The remaining problem of solving (4) is how to get the initial latent image, i.e.  $\mathcal{I}_T$ . We use the fact that the blurred

image  $\bar{\mathcal{I}}$  roughly equals the average of the instant images in the exposure process. Combining this fact with (6), we have

$$\bar{\mathcal{I}} \approx \frac{1}{T} \sum_{i=1}^T \mathcal{I}_i = \mathcal{I}_T \odot \frac{1}{T} \left( 1 + \sum_{t=2}^T \prod_{i=1}^{t-1} \mathcal{B}_{T-i+1}^{T-i} \right), \quad (8)$$

where  $\mathcal{B}_{i+1}^i = \exp(-\tau \mathcal{S}_{i+1}^i)$  and  $\mathcal{S}_{i+1}^i$  is defined in (6). It provides an initial estimation of  $\mathcal{I}_T$ , namely  $\hat{\mathcal{I}}_T$ , using the blurred image  $\bar{\mathcal{I}}$  and events. Thus, we also treat solving  $\mathcal{I}_T$  a denoising problem, centered around  $\hat{\mathcal{I}}_T$ , and use a network to approximate it. We note, however, the accumulative operator in (8) introduces more drift unlike the sequential deblurring steps. We thus correct  $\hat{\mathcal{I}}_T$  via a separate and more powerful network:  $\mathcal{I}_T^* = \mathcal{N}_0(\hat{\mathcal{I}}_T, \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T})$ . The full deblurring process is summarized in Alg. 1. Note that by design (7), the latent image is conditioned on both local and long-term cues from the image and events.

---

#### Algorithm 1 Event-assisted Deep Motion Deblurring

---

**Require:** the blurred image  $\bar{\mathcal{I}}$ , events  $\mathbb{E}_{1 \sim T}$

- 1: Get initial estimate  $\hat{\mathcal{I}}_T$  by solving (8)
  - 2: Deblurring:  $\mathcal{I}_T^* = \mathcal{N}_0(\hat{\mathcal{I}}_T, \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T})$
  - 3: Initialize counter:  $i = T - 1$
  - 4: **while**  $i \geq 1$  **do**
  - 5:   Get initial estimate  $\hat{\mathcal{I}}_i$  by solving (2)
  - 6:   Deblurring:  $\mathcal{I}_i^* = \mathcal{N}(\hat{\mathcal{I}}_i, \mathcal{I}_{i+1}, \mathbb{E}_{i \sim i+1}, \bar{\mathcal{I}}, \mathbb{E}_{1 \sim T})$
  - 7:    $i \leftarrow i - 1$
  - 8: **end while**
  - 9: **return** Deblurred sequence  $\mathbb{I}^* = \{\mathcal{I}_i\}_{i=1}^T$
- 

## 4. Network Architecture

Fig. 2 shows the proposed event-based motion deblurring architecture, which contains: a *read* network that traverses over the events and generates a single representation of the global scene motion, an *initialize* network that couples appearance and motion to generate the initial latent image, and the recurrent *process* network sequentially deblurring all the latent images<sup>1</sup>. The read and initialize networks instantiates  $\mathcal{N}_0$  while the process network implements  $\mathcal{N}$  in Alg. 1.

The read network reads all the event data and generate a joint representation that accounts for the global event motion. To accomplish that, events during the exposure are first binned into equal-length time intervals (3 intervals in Fig. 2). In each time interval, events are represented with stacked event frames [17], through further dividing an interval into 8 equal-size chunks, summing over the polarities of events falling into each chunk, and stacking the results along channel dimension. The read network is a recurrent

<sup>1</sup>Due to space limit we briefly describe the component design and refer the detailed layer/parameter configurations to our supplementary material.

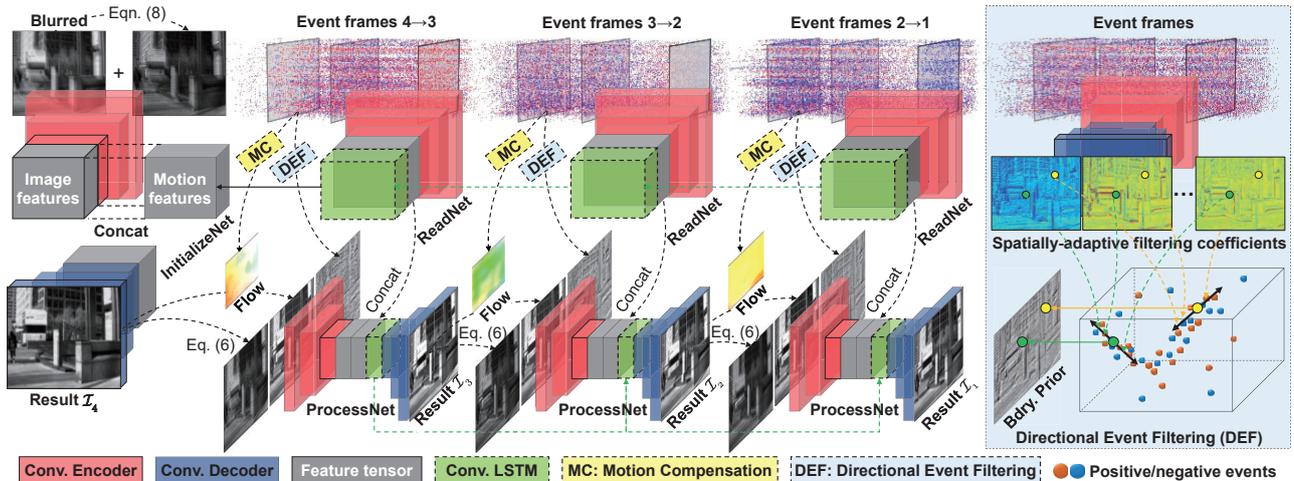


Figure 2. The proposed learning framework for event-based motion deblurring. For better visualization, we only assume 4 sharp frames are recovered from the blurred image. Detailed layer and parameter configurations are referred to the supplementary material. Note that the Motion Compensation (MC) module is not illustrated due to the lack of space. See text for detailed description of the architecture.

encoder consisting of convolutional blocks and a convolutional LSTM [41] on top to accumulate features along time.

The initialize network decodes the appearance from the blurred image and couples it with the global motion to solve the latent image  $\mathcal{I}_T^*$ . It takes as input both the blurred image  $\tilde{\mathcal{I}}$  and the initial estimate  $\hat{\mathcal{I}}_T$  (via solving Eqn. (8)) and processes them with a convolutional encoder, concatenates the encodings with the accumulated global motion features from the read network, and feeds the joint features into a decoder to get the result.

Given the initial result, the process network then sequentially deblurs the remaining latent images. In the  $i$ th step, it consumes both image and event-based observations. The image part include: 1) the initial estimate  $\hat{\mathcal{I}}_i$  as obtained by Eqn. (6) using the previous reconstruction  $\mathcal{I}_{i+1}$ , 2) the local historical image by transforming the previous result  $\mathcal{I}_{i+1}$  with the *Motion Compensation* (“MC” in Fig. 2) module, and 3) the boundary guidance map given by the *Directional Event Filtering* (“DEF” in Fig. 2) module. These two modules will be explained further shortly after. Input images are processed by convolutional layers and concatenated with the per-step event features extracted from the read network via latent fusion. The fused features are processed and fed to another convolutional LSTM to propagate temporal knowledge along time. Finally, a decoder takes the joint features and generates the deblurred image.

**Motion compensation.** We use a motion compensation module to warp previous deblurring result  $\mathcal{I}_{i+1}$  and generate an initialization of the  $i$ th time step. Although Eqn. (6) achieves this by event integration, we find it more effective to predict a flow field from which we directly warp the clean result  $\mathcal{I}_{i+1}$  as additional guidance. Motion compensation

for events have already been discussed in [10]. For efficiency, we adopt a FlowNetS architecture [9] to take events  $\mathbb{E}_{i \sim i+1}$  as input and directly regress forward flows from  $i$  to  $i + 1$ . Warping is implemented with a differentiable spatial transformer layer [18, 14].

**Directional event filtering.** The initial estimates  $\hat{\mathcal{I}}_i$  may suffer unaddressed blur due to the naive blurring model (8) and the noisiness of events. We alleviate this issue with the aid of sharp boundary prior, a widely explored image prior for blind deblurring [7, 46], extracted from events  $\mathbb{E}_{i \sim i+1}$ .

Events indicate local change of scene illuminance and reveal physical boundaries. However, as scene boundaries are moving, at a specific time they are only spatially aligned with the *latest* events triggered at their positions. As a toy example, Fig. (3) shows after the imaging the top and bottom lines correspond to events at two different time points. It gives that one can generate scene boundary prior by sampling events at proper space-time positions. Note that due to variation of scene depth, different scene parts may have distinct motion, and position-adaptive sampling is essential.

Besides, as events are sparse, noisy, and non-uniformly distributed signals, a robust sampling process should decide both where (*i.e.* center) and how many (*i.e.* scale) to sample. We learn this task from data via differentiable sampling and filtering. For each image position  $\mathbf{p}$ , a temporal center  $c(\mathbf{p})$  and a set of  $2k + 1$  filtering coefficients  $\{\alpha_i\}_{i=-k}^k$ , where  $k$  is the support of filtering kernel, are predicted with a small network from the events, satisfying  $\forall i, \alpha_i \geq 0$  and  $\sum_{i=-k}^k \alpha_i = 1$ . The filtered result is obtained by

$$\mathcal{G}(\mathbf{p}) = \sum_{i=-k}^k \alpha_i s(\mathbf{p} + \lambda k \mathbf{d}(\mathbf{p}, c(\mathbf{p})), c(\mathbf{p}) + \lambda k), \quad (9)$$

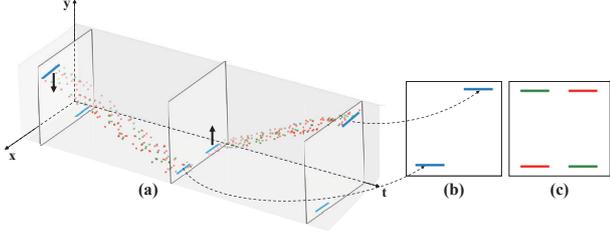


Figure 3. Motivation of adaptive event sampling. (a) A toy scene where the top line moves down first, after which the bottom line moves up. Events with positive and negative polarities are shown as red and green dots, respectively. (b) The projected image of the scene after the imaging process. Scene boundaries correspond to the latest triggered events, which may vary for different positions, as indicated by arrows. (c) The accumulation map of events.

where  $\lambda$  defines sampling stride (we use  $k = 2, \lambda = 1$ ),  $s(\cdot, \cdot)$  denotes a sampling function in space-time domain. For the stacked event frame representation of events  $\mathbb{E}_i^{z+1}$ , one can apply the trilinear kernel for continuous sampling [21]. Note that the velocity  $\mathbf{d}$  should follow the direction of local motion of events at space-time point  $(\mathbf{p}, c(\mathbf{p}))$  to filter along the density surface of events but not across it.

To get local velocity, we reuse the flow vectors predicted by motion compensation module. We assume object velocity stays constant, which is roughly true in this context as there is just a fraction of time duration (*i.e.* only  $1/(T-1)$  the exposure). Motion compensation gives the velocities of all the positions  $\mathbf{p}_0 \in \mathbb{P}$  at time  $i$ ,  $\mathbf{d}(\mathbf{p}_0, i)$ . At time  $c(\mathbf{p})$ , a pixel  $\mathbf{p}_0$  would be shifted by the flows to a new position:

$$\mathbf{n}(\mathbf{p}_0) = \mathbf{p}_0 + (c(\mathbf{p}) - i) \mathbf{d}(\mathbf{p}_0, i). \quad (10)$$

Note that  $\mathbf{n}(\mathbf{p}_0)$  inherits the velocity of  $\mathbf{p}_0$  under the local constancy assumption:  $\mathbf{d}(\mathbf{n}(\mathbf{p}_0), c(\mathbf{p})) = \mathbf{d}(\mathbf{p}_0, i)$ .

However, the intersected positions at time plane  $c(\mathbf{p})$ , namely  $\{\mathbf{n}(\mathbf{p}_0) | \mathbf{p}_0 \in \mathbb{P}\}$ , does not ensure complete sampling of the image space. Thus, we resample the velocity at a given target  $\mathbf{p}$  with a Nadaraya-Watson estimator [4]:

$$\mathbf{d}(\mathbf{p}, c(\mathbf{p})) = \frac{\sum_{\mathbf{p}_0 \in \mathbb{P}} \kappa(\mathbf{n}(\mathbf{p}_0) - \mathbf{p}) \mathbf{d}(\mathbf{n}(\mathbf{p}_0), c(\mathbf{p}))}{\sum_{\mathbf{p}_0 \in \mathbb{P}} \kappa(\mathbf{n}(\mathbf{p}_0) - \mathbf{p})}, \quad (11)$$

where the kernel  $\kappa$  is simply defined with a standard Gaussian. This in spirit shares similarity with the ‘gather’ approach in computer graphics for surface rendering [49].

Eqn. (11) uses all  $\mathbf{p}_0$ s to estimate each position  $\mathbf{p}$ , which is inefficient. In practice we only use samples located within a local  $L \times L$  window centered around  $\mathbf{p}$ . The window size  $L$  should account for the maximal spatial displacement of pixels, which we find  $L = 20$  sufficient. All of the proposed steps are differentiable, and can be plugged into the network for end-to-end training.

**Loss Function.** We use the following joint loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{content} + \lambda_a \mathcal{L}_{adv} + \mathcal{L}_{flow} + \lambda_t \mathcal{L}_{tv}, \quad (12)$$

Here,  $\mathcal{L}_{content}$  is the photometric  $\ell_1$  loss  $\frac{1}{T} \sum_{i=1}^T \|\mathcal{I}_i^* - \mathcal{I}_i^g\|$ , where  $\mathcal{I}_i^g$  is the groundtruth clean image. To improve sharpness of the result, we also incorporate an adversarial loss  $\mathcal{L}_{adv}$ . We use the same PatchGAN discriminator [13] and follow its original loss definitions strictly.

The flow network introduces two other loss terms. The first  $\mathcal{L}_{flow}$  is the photometric reconstruction loss:

$$\mathcal{L}_{flow} = \frac{1}{T-1} \sum_{i=1}^{T-1} \|\omega(\mathcal{I}_{i+1}^*, \mathcal{F}_{i \rightarrow i+1}) - \mathcal{I}_i^g\|, \quad (13)$$

where  $\omega(\cdot, \cdot)$  is a backward warping function using forward flows  $\mathcal{F}_{i \rightarrow i+1}$ , and  $\mathcal{L}_{tv} = \frac{1}{T-1} \sum_{i=1}^{T-1} \|\nabla \mathcal{F}_{i \rightarrow i+1}\|$  is the total variation loss for flow field smoothing. For these terms, we follow the same definitions of [14]. The weights  $\lambda_a$  and  $\lambda_t$  are set to 0.01 and 0.05, respectively.

## 5. Experiments

### 5.1. Experimental Settings

**Dataset preparation.** We use two datasets for evaluation. First, we evaluate on the GoPro [25] dataset which is widely adopted for image motion deblurring and recently used by [31] to benchmark event-based deblurring. To synthesize events reliably, we use the open ESIM event simulator [35]. We follow the suggested training and testing split. The blurred image is also provided officially by averaging nearby (the number varies from 7 to 13) frames.

As there lacks a large-scale dataset for evaluating event-based motion deblurring in real-world scenarios, we capture a novel dataset of urban environment, called *Blur-DVS*, with a DAVIS240C camera. It hybrids a high speed event sensor with a low frame-rate Active Pixel Sensor (APS) recording intensity images at  $180 \times 240$ . Thus, APS may suffer motion blur in fast moving. We collect two subsets for evaluation. The *slow* subset consists of 15246 images captured with slow and stable camera movement of relatively static scenes, thus motion blur rarely happens. We synthesize motion blurs by averaging nearby 7 frames, resulting into 2178 pairs with blurred image and sharp sequence. In this manner, we can conduct quantitative benchmarkings. We select 1782 pairs for training, and 396 for testing. The *fast* subset consists of additional 8 sequences with 740 frames in total, captured under fast camera movement of fast moving scenes to investigate how the proposed approach generalizes to real motion blur. However, there is no groundtruth data available on this subset.

**Method comparison.** We conduct extensive comparisons with recent motion deblurring methods with available

Table 1. Single-image motion deblurring performance on the GoPro dataset.

Models	DCP [29]	MBR [42]	FLO [11]	EVS [15]	SRN [43]	SVR [52]	DMS [25]	MPN [50]	BHA [31]	Ours
PSNR	23.50	25.30	26.05	26.98	30.26	29.18	29.08	31.50	29.06	<b>31.79</b>
SSIM	0.834	0.851	0.863	0.892	0.934	0.931	0.914	0.948	0.943	<b>0.949</b>



Figure 4. Visual comparisons on the GoPro dataset. From left to right, we show two examples with the blurred image, results of MPN [50], BHA [31] and our approach, as well as groundtruth sharp image, respectively. Zoom in for better view.

Table 2. Video reconstruction performance on the GoPro dataset.

Models	CIE [38]	CIE+SRN*	EVS [15]	BHA [31]	Ours
PSNR	25.84	26.34	25.62	28.49	<b>29.67</b>
SSIM	0.790	0.809	0.856	0.920	<b>0.927</b>

\*A hybrid baseline that adopts CIE to reconstruct images first, then SRN to deblur each image. See [31] for details.

results and/or codes. They include image-based methods: DCP [29], MBR [42], FLO [11], DMS [25], EVS [15], SRN [43], SVR [52] and MPN [50], and the state-of-the-art event-based motion deblurring method BHA [31]. We also compare with three event-based video reconstruction methods, including CIE [38], MRL [24] and the state-of-the-art learning-based approach ETV [36]. PSNR and SSIM metrics are used for quantitative evaluation.

**Implementation details.** For both datasets, our training adopts a batch size of 2 training pairs and Adam optimizer. The network is trained for 400 epochs, with a learning rate  $10^{-4}$  at the beginning and linearly decayed to zero starting from the 200th epoch. All the components of the network are jointly trained from scratch.

## 5.2. Comparisons with State-of-the-Art Models

On the GoPro dataset, we report the results on both single image deblurring (*i.e.* only recovering the middle frame) and video reconstruction (*i.e.* recover all the sharp frames) in Table 1 and 2, respectively. Numbers of other approaches are directly taken from papers. Our approach achieves the top place in both tasks, demonstrating the advantages of event-assisted deblurring than purely relying on images, and the superiority of the proposed framework over physical reconstruction model. We show visual comparisons on two fast moving scenes in Fig. 4: while image-based method MPN cannot well address such blur, BHA is sensitive to the noise of events especially along object edges. Our approach

generates cleaner and sharper results.

Note that GoPro dataset mainly presents small to moderate motion blur, thus the blurred input is of good quality and improvement from events is marginal. Thus recent powerful architectures SRN and MPN get very promising results though they do not see events. For this reason, we compare our approach with state-of-the-art methods on the proposed Blur-DVS dataset, in which severe motion blur are more universal. Again, we report results on single image deblurring (Table 3) and video reconstruction (Table 4) tasks. Note that for fair comparisons, The learning-based methods SRN, MPN and ETV are finetuned on the training set of Blur-DVS. We also compare with their enhanced versions that see both image and events: for image-based methods SRN and MPN, we concatenate the input blurred image with all the 48 (8 binned frames in each time interval and  $(7 - 1)$  intervals) event frames. For the event-based method ETV, we also feed the blurred image along with the events to each of its recurrent reconstruction step. We denote these variants as SRN+, MPN+ and ETV+, respectively.

In Table 3 and 4, the proposed approach achieves the best results. It also outperforms all the enhanced variants, demonstrating the effectiveness of the proposed framework. Fig. 5 illustrates that: 1) in case of fast motion, image-based cues alone are not sufficient, limiting performance of MPN; 2) the physical model BHA is prone to noise and presents unaddressed blur due to the lossy sampling mechanism of events; 3) event-based reconstruction methods CIE, MRL and ETV do not restore scene contrast correctly due to the lack of image guidance and/or the simplified physical model. Our approach does not suffer the mentioned issues, and presents sharper results even than the enhanced image+event variants equipped with powerful architectures.

Finally, we analyse the generalization behavior to real-world motion blur. As shown in Fig. 6, the proposed ap-

Table 3. Single-image deblurring performance on the Blur-DVS dataset.

Models	DMS [25]	SRN [43]	SRN+*	MPN [50]	MPN+*	CIE [38]	MRL [24]	ETV [36]	ETV+*	BHA [31]	Ours
PSNR	20.48	20.21	24.92	23.52	26.08	19.02	10.59	16.89	24.81	22.43	<b>26.48</b>
SSIM	0.572	0.567	0.821	0.753	0.831	0.478	0.195	0.597	0.790	0.715	<b>0.839</b>

\*SRN+, MPN+ and ETV+ denote enhanced versions of SRN, MPN, ETV respectively. See text for details.

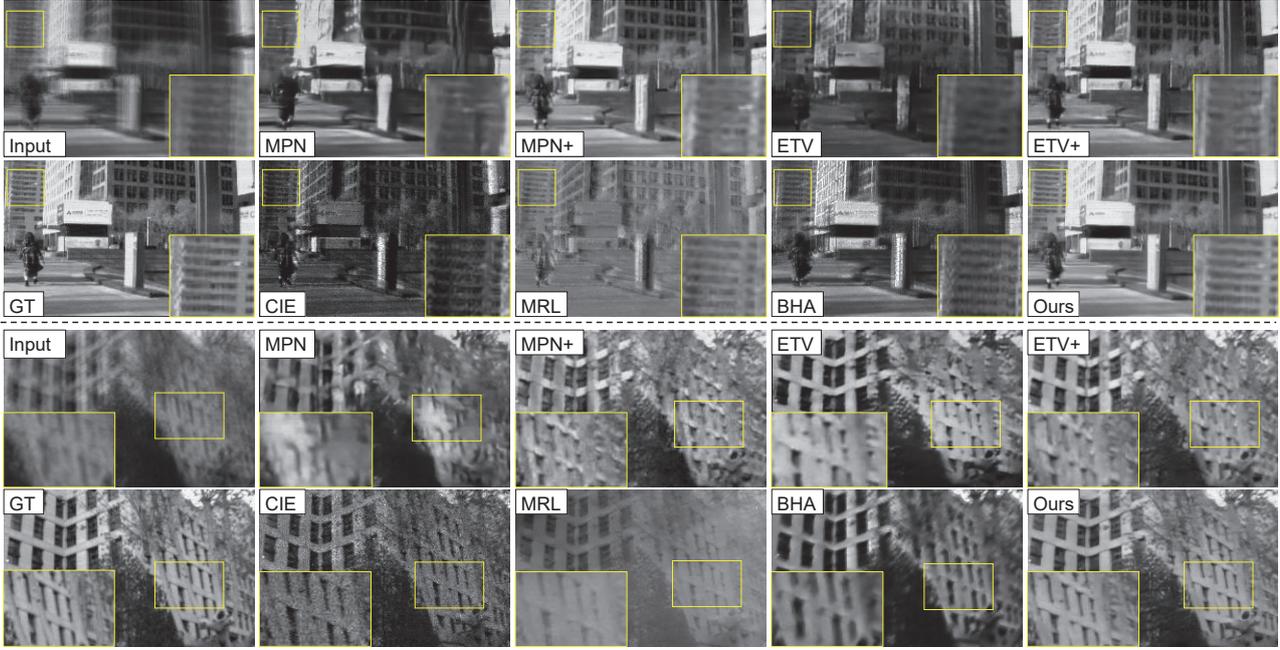


Figure 5. Representative results of two examples generated by different approaches on the *slow* subset of Blur-DVS dataset. More results can be found in our supplementary material. Zoom in for better view.

Table 4. Video reconstruction performance on Blur-DVS dataset.

Models	CIE [38]	MRL [24]	ETV [36]	ETV+	BHA [31]	Ours
PSNR	18.94	10.57	16.60	24.10	22.06	<b>25.33</b>
SSIM	0.473	0.194	0.587	0.777	0.699	<b>0.827</b>

proach achieves the best visual quality. We suspect that the explicit modeling of motion deblurring and introduction of strong deblurring priors may alleviate the learning difficulty and avoid potential overfitting in more black-box architectures. In practice we find such improvement consistent on real data, as demonstrated by more results on the *fast* subset provided in our supplementary material.

### 5.3. Performance Analysis

**Analysing different components.** We isolate the important algorithm components to see their contributions to the final performance, and summarize the results in Table 5 and Fig. 7. As it shows, each component is necessary to improve the PSNR and SSIM of the results. Using image appearance only without events (App.) cannot deblur the image well. Using events only, on the other hand, recovers plenty of details but intensity contrast is not well recovered (see Fig. 7

Table 5. Component analysis on the Blur-DVS dataset. “App.” and “event” denotes using the blurred image appearance and event data as input, respectively. “MC” and “DEF” refer to the motion compensation and directional event filtering modules, respectively.

App.	Event	+MC	+DEF	PSNR	SSIM
✓	✗	✗	✗	16.50	0.418
✗	✓	✓	✗	16.38	0.560
✓	✓	✗	✗	23.39	0.760
✓	✓	✓	✗	24.71	0.786
✓	✓	✓	✓	25.33	0.827

(b)). Using both input signals (App. + event) achieves better results, but the reconstructed image is not very smooth due to noise (e.g. the ground in Fig. 7 (c)). Further incorporating motion compensation (+MC) helps in these aspects as it imposes temporal smoothness. Finally, further introducing the directional event filtering module (+DEF), sharper results and richer details can be generated thanks to the learned boundary guidance.

**Justification of the DEF module.** In Table 6, we justify the necessity of the proposed directional event filtering module. Here, “w/o guid.” does not include boundary guidance in the whole pipeline. On the contrary, “guid

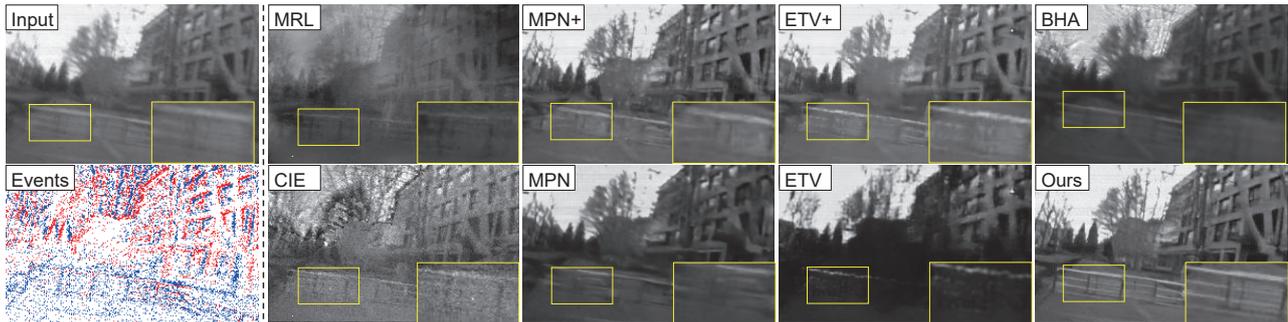


Figure 6. Representative results generated by different approaches on the *fast* subset (real-world motion blur) of Blur-DVS dataset. More results can be found in our supplementary material. Zoom in for better view.

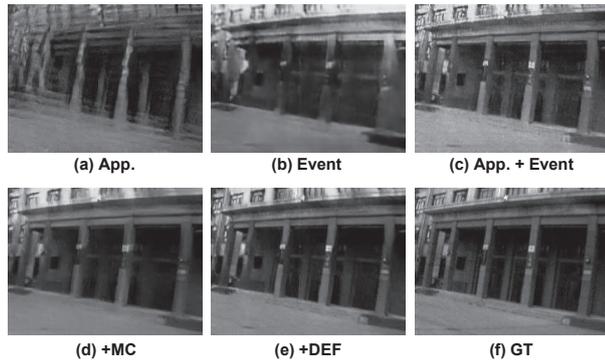


Figure 7. Visually analysing the contributions of different components on the DVS-Blur dataset. See text for details.

Table 6. Analysing the directional event filtering module on the DVS-Blur dataset. See text for details.

Models	guid. only	w/o guid.	full	+param.
PSNR	25.16	24.71	25.33	24.64
SSIM	0.816	0.786	0.827	0.788

only.” discards event features in each sequential deblurring step while using boundary guidance only as additional cue. We further design a variant “+param.”, which does not incorporate DEF but has additional convolution layers in the encoder of *process* network which exceeds the current parameter size. Results show that the learned boundary guidance greatly improves the estimation (from 0.786 to 0.827 in SSIM), and itself without other cues can already leads to promising results. Simply enlarging the network size, however, does not observe meaningful improvement.

In Fig. 8, we visualize the impact of learned boundary guidance. Note how the network learns to select different time centers according to the scene’s motion (Fig. 8 (c)). Boundary guidance improves the sharpness of the scene significantly and recovers missing details (Fig. 8 (e) and (f)).

**Low-light photography.** A potential application of the proposed approach is low-light photography, as shown in Fig 9. The short-exposure (13ms) image is light-starved. The long-exposure (104ms) one, however, may suffer severe motion blur. Leveraging event cues, our approach generates natural results without such blur.

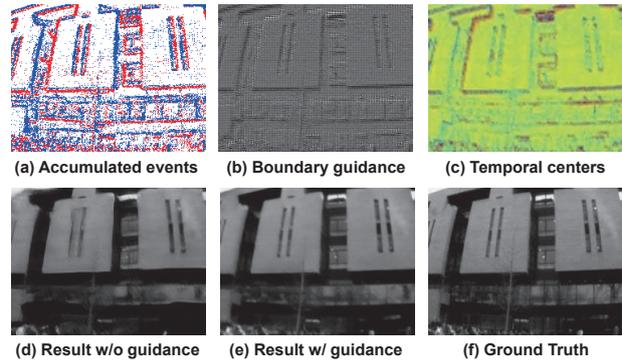


Figure 8. Visualizing learned boundary guidance. Note how motion boundaries from different time stamps are selected in the attention map (c) (red for large value and blue for small values).

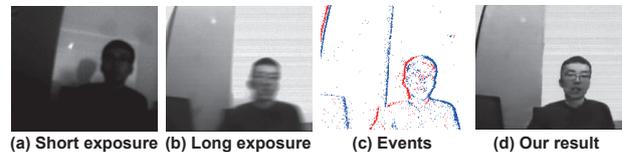


Figure 9. Low-light photography using our approach. Images and events are captured with DAVIS240C camera in an indoor scene.

## 6. Conclusion

In this work, we propose to extract a video from a severe motion-blurred image under the assistance of events. To this end, a novel deep learning architecture is proposed to effectively fuse appearance and motion cues at both global and local granularity. Furthermore, sharp event boundary guidance is extracted to improve reconstructed details with a novel directional event filtering module. Extensive evaluations show that the proposed approach achieves superior performance than various existing image and event-based methods, on both synthetic and real-world datasets.

**Acknowledgements.** We thank the reviewers for their valuable feedback. This work is supported by Beijing Posdoctoral Research Foundation (Grant No. ZZ-2019-89), National Key R&D Program of China under contract No. 2017YFB1002201, National Natural Science Fund for Distinguished Young Scholar (Grant No. 61625204) and partially supported by the Key Program of National Science Foundation of China (Grant No. 61836006).

## References

- [1] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner. A low power, high throughput, fully event-based stereo system. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7532–7542, 2018.
- [2] Y. Bahat, N. Efrat, and M. Irani. Non-uniform blind deblurring by reblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3306–3314, 2017.
- [3] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016.
- [4] H. J. Bierens. The Nadaraya-Watson Kernel regression function estimator. *Serie Research Memoranda*, (0058), 1988.
- [5] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbrück. A  $240 \times 180$  130 db 3  $\mu$ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.
- [6] A. Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision (ECCV)*, pages 221–235, 2016.
- [7] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5):145, 2009.
- [8] J. Dong, J. Pan, Z. Su, and M.-H. Yang. Blind image deblurring with outlier handling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2497–2505, 2017.
- [9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.
- [10] G. Gallego, H. Rebecq, and D. Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2018.
- [11] D. Gong, J. Yang, L. Liu, Y. Zhang, I. D. Reid, C. Shen, A. Hengel, and Q. Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3806–3815, 2017.
- [12] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *IEEE International Conference on Computer Vision (ICCV)*, pages 463–470, 2011.
- [13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.
- [14] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. G. Learned-Miller, and J. Kautz. Super slo-mo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018.
- [15] M. Jin, G. Meishvili, and P. Favaro. Learning to extract a video sequence from a single motion-blurred image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6334–6342, 2018.
- [16] T. H. Kim, B. Ahn, and K. M. Lee. Dynamic scene deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3160–3167, 2013.
- [17] Y.-S. Ho L. Wang, S. M. Mostafavi and K.-J. Yoon. Event-based High Dynamic Range Image and Very High Frame Rate Video Generation using Conditional Generative Adversarial Networks. 2019.
- [18] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang. Learning blind video temporal consistency. In *European Conference on Computer Vision*, pages 179–195, 2018.
- [19] P. Lichtsteiner, C. Posch, and T. Delbrück. A  $128 \times 128$  120 db 15  $\mu$ s latency asynchronous temporal contrast vision sensor. *Journal of Solid-State Circuits*, 43(2):566–576, 2008.
- [20] M. Liu and T. Delbrück. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Machine Vision Conference (BMVC)*, page 88, 2018.
- [21] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4473–4481, 2017.
- [22] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision (ECCV)*, pages 783–798, 2014.
- [23] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos. Event-based moving object detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.
- [24] G. Munda, C. Reinbacher, and T. Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision (IJCV)*, 126(12):1381–1393, 2018.
- [25] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 257–265, 2017.
- [26] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4762–4770, 2017.
- [27] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *German Conference on Pattern Recognition (GCPR)*, pages 65–77, 2017.
- [28] J. Pan, J. Dong, Y.-W. Tai, Z. Su, and M.-H. Yang. Learning discriminative data fitting functions for blind image deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1077–1085, 2017.
- [29] J.-S. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1628–1636, 2016.
- [30] L. Pan, Y. Dai, M. Liu, and F. Porikli. Simultaneous stereo video deblurring and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6987–6996, 2017.

- [31] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [32] H. Park and K. M. Lee. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4623–4631, 2017.
- [33] S. Pini, G. Borghi, R. Vezzani, and R. Cucchiara. Learn to See by Events: RGB Frame Synthesis from Event Cameras. *ArXiv preprints arXiv:1812.02041 [cs.CV]*, 2018.
- [34] B. Ramesh, S. Zhang, Z. W. Lee, Z. Gao, G. Orchard, and C. Xiang. Long-term object tracking with a moving event camera. In *British Machine Vision Conference (BMVC)*, page 241, 2018.
- [35] H. Rebecq, D. Gehrig, and D. Scaramuzza. ESIM: an open event camera simulator. *Conference on Robotics Learning (CoRL)*, 2018.
- [36] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [37] W. Ren, J. Pan, X. Cao, and M.-H. Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1086–1094, 2017.
- [38] C. Scheerlinck, N. Barnes, and R. Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision (ACCV)*, 2018.
- [39] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 604–611, 2013.
- [40] P. A. Shedligeri and K. Mitra. Photorealistic Image Reconstruction from Hybrid Intensity and Event based Sensor. *ArXiv preprint arXiv:1805.0614 [cs.CV]*, 2018.
- [41] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems (NIPS)*, pages 802–810, 2015.
- [42] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 769–777, 2015.
- [43] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018.
- [44] P. Wieschollek, M. Hirsch, B. Schölkopf, and H. P. A. Lensch. Learning blind motion deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 231–240, 2017.
- [45] L. Xiao, J. Wang, W. Heidrich, and M. Hirsch. Learning high-order filters for efficient blind deconvolution of document photographs. In *European Conference on Computer Vision*, pages 734–749, 2016.
- [46] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, pages 157–170, 2010.
- [47] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6978–6986, 2017.
- [48] C. Ye, A. Mitrokhin, C. Fermüller, J. A. Yorke, and Y. Aloimonos. Unsupervised Learning of Dense Optical Flow, Depth and Egomotion from Sparse Event Data. *ArXiv preprints arXiv:1809.08625 [cs.CV]*, 2018.
- [49] J. Yu and G. Turk. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Transactions on Graphics*, 32(1):5:1–5:12, 2013.
- [50] H. Zhang, Y. Dai, H. Li, and P. Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5978–5986, 2019.
- [51] H. Zhang and D. P. Wipf. Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1556–1564, 2013.
- [52] J. Zhang, J. Pan, J. S. J. Ren, Y. Song, L. Bao, R. W. H. Lau, and M.-H. Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2521–2529, 2018.
- [53] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, 2017.
- [54] A. Z. Zhu, Y. Chen, and K. Daniilidis. Realtime time synchronized event-based stereo. In *European Conference on Computer Vision (ECCV)*, pages 438–452, 2018.
- [55] W. Zuo, D. Ren, S. Gu, L. Lin, and L. Zhang. Discriminative learning of iteration-wise priors for blind deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3232–3240, 2015.