DeblurSR: Event-Based Motion Deblurring under the Spiking Representation

Chen Song, Chandrajit Bajaj, Qixing Huang
The University of Texas at Austin, Austin, TX 78712, USA
{song, bajaj, huangqx}@cs.utexas.edu

Abstract

We present DeblurSR, a novel motion deblurring approach that converts a blurry image into a sharp video. DeblurSR utilizes event data to compensate for motion ambiguities and exploits the spiking representation to parameterize the sharp output video as a mapping from time to intensity. Our key contribution, the Spiking Representation (SR), is inspired by the neuromorphic principles determining how biological neurons communicate with each other in living organisms. We discuss why the spikes can represent sharp edges and how the spiking parameters are interpreted from the neuromorphic perspective. DeblurSR has higher output quality and requires fewer computing resources than state-of-the-art event-based motion deblurring methods. We additionally show that our approach easily extends to video super-resolution when combined with recent advances in implicit neural representation.

Introduction

Neuromorphic events are commonly used in deblurring algorithms (Pan et al. 2019, 2020; Jiang et al. 2020; Lin et al. 2020; Wang et al. 2020; Shang et al. 2021; Zhang et al. 2021a; Han et al. 2021; Xu et al. 2021; Sun et al. 2022; Kim et al. 2021; Song, Huang, and Bajaj 2022; Wang et al. 2019). Modern neuromorphic devices are extremely fast and capture up to one million events per second. The enormous density of neuromorphic events has been shown to enable motion-deblurring algorithms to reverse the exposure process and recover the relative movement between the camera and the environment from one single image (Pan et al. 2019; Wang et al. 2020; Song, Huang, and Bajaj 2022). Figure 1 presents an illustration of the event-based image-to-video motion deblurring task.

While early motion-deblurring works apply numerical optimization techniques to directly solve for the sharp output video (Pan et al. 2019, 2020; Wang et al. 2019), recent approaches utilize different data-driven pipelines as the inference model (Wang et al. 2020; Song, Huang, and Bajaj 2022). Despite these attempts, it is yet unclear how to properly instill prior knowledge about the neuromorphic event mechanism to build an effective deep learning paradigm that simultaneously addresses the motion ambiguity and emphasizes sharp edges.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
Figure 2: The Spiking Representation. Let $L_{xy}(t)$ be the intensity of pixel $(x, y)$ at time $t$. $L_{xy}(t)$ is approximated by a piece-wise linear function, analogous to the membrane potential of the neuron on the retina dedicated to pixel $(x, y)$. 

1. The membrane potential does not change when the neuron is at rest. The pixel’s potential stays constant in the absence of motion. 
2. Spikes occur at segment endpoints. The gap resembles the strength of the spike, which captures edge features caused by instantaneous changes. 
3. The slope simulates the rate of membrane potential leakage in a biological neuron after its excitement, enabling the spiking representation to model gradual intensity changes. While the spiking representation (orange) captures signal discontinuities, the continuous parameterization (blue) does not.

As shown in Figure 2, we use a novel Spiking Representation (SR) to approximate the per-pixel mappings from time to intensity. Imagine there is a dedicated visual receptor neuron for each pixel on the human retina. We model the light intensity of each pixel as the membrane potential of the corresponding receptor neuron, defined as the voltage difference between the interior and the exterior of the neuron cell. The spiking representation exploits a piece-wise linear function similar to the one developed by neuroscientists in the 1900s under the name of the Leaky-Integrate-and-Fire model (Lapicque 1907). Each linear segment’s slope characterizes the gradual intensity change rate over time. The gaps at segment endpoints represent abrupt intensity changes caused by the edge movements. The spiking parameters, including the slope and intercept of each line segment and the segment endpoints, are predicted by a deep neural network. We further integrate the per-pixel piece-wise linear parameterization with spatial convolutional kernels, encouraging information to propagate among neighboring pixels. We refer to the proposed intensity modeling as the spiking representation because it mimics the spiking mechanism in biological neurons.

We conduct extensive experimental analysis on the REDS dataset (Nah et al. 2019) with synthetic events and the HQF dataset (Stoffregen et al. 2020) with real events. DeblurSR improves state-of-the-art reconstruction quality by 12.3% on REDS and 22.2% on HQF, while requiring a shorter training time and fewer computing resources. Through ablation studies, we demonstrate the strengths of three optional components, including the automatic keypoint selection module, the integral normalization constant, and the kernel-based spatial modeling. Finally, we show how DeblurSR naturally extends to video super resolution, improving the state-of-the-art method by 7.8%.

In summary, we make the following contributions:

- We propose a novel Spiking Representation (SR) to parameterize sharp videos. We discuss the similarity between the spikes and the biological neural mechanism. We explain how the gaps in the spiking representation can be used to capture sharp edges.
- We train a deep network that regresses the spiking parameters from a blurry image and its associated events during the exposure interval. We show how to use the predicted parameters to render sharp videos.
- In addition to motion deblurring, we extend the spiking representation to support video super resolution.

**Related Work**

**Event Cameras and Spiking Neural Networks**

Event cameras (Lichtsteiner, Posch, and Delbruck 2008; Mahowald 1992) react to intensity changes and produce a stream of events. Each event is represented as a 4-tuple $(x, y, t, p)$, including the pixel coordinates, the timestamp, and the binary polarity of the intensity change. Modern event cameras are able to detect up to one million events per second, allowing them to capture extremely fast motion details. However, events are noisy and unable to account for the exact magnitudes of the intensity changes, leading to challenges in algorithm design. A thorough introduction to the technical properties of event cameras is beyond the scope of this paper. We refer interested readers to Gallego et al. (2020) for a comprehensive survey.

Spiking Neural Networks (SNNs) (Maass 1997) are popular tools to build an event-based vision system because both event cameras and SNNs simulate how biological neurons behave. Different mathematical models are constructed to explain the spiking mechanism (Lapicque 1907; Dutta et al. 2017; Hodgkin and Huxley 1952; Izhikevich 2003; Borisyuk and Borisuyk 1997). Our work is particularly inspired by the Leaky-Integrate-and-Fire (LIF) model (Lapicque 1907), where each neuron has a membrane potential whose value decays linearly with time in the logarithmic space and plunges abruptly whenever there is a spike. While prior works in event vision exploit SNNs as a black-box inference network (O’Connor et al. 2013; Diehl et al. 2015; Esser et al. 2016; Rueckauer et al. 2017; Zhang et al. 2021b; Zhu et al. 2022; Zhang et al. 2022; Lee, Kosta, and Roy 2022), we extend the LIF model to directly model intensity changes in a video under the spiking representation. For readers familiar with SNNs, the most significant difference between SNNs and DeblurSR is that SNNs implicitly store the membrane potential and explicitly output the spikes, whereas DeblurSR explicitly models the membrane potential of visual receptor neurons and implicitly represents the spikes through parameterized coefficients.
Motion Deblurring

Motion deblurring algorithms turn a blurry image into a sharp video, allowing humans and downstream vision applications to understand movements during the image formation process. Motion deblurring is an ill-posed problem because the blurry image alone fails to capture critical motion parameters such as the moving direction and speed (Jin, Meishvili, and Favaro 2018; Purohit, Shah, and Rajagopalan 2019). In Figure 1, it is impossible to tell whether the photographer is moving forward into the woods or backward out of the screen from the blurry image itself. Thanks to the fast data rate of event cameras, several prior works utilize event streams to supplement the blurry input image to address the motion ambiguity (Pan et al. 2019, 2020; Jiang et al. 2020; Lin et al. 2020; Wang et al. 2020; Shang et al. 2021; Zhang et al. 2021a; Han et al. 2021; Xu et al. 2021; Tulyakov et al. 2021; Paikin et al. 2021; Tulyakov et al. 2022; Zhang and Yu 2022; Song, Huang, and Bajaj 2022; Sun et al. 2022; Kim et al. 2021; Wang et al. 2019) in not only image-to-video deblurring but also image-to-image deblurring and frame interpolation. Closely related to our design, E-CIR (Song, Huang, and Bajaj 2022) represents pixel intensities in the output video as polynomial functions characterized by the temporal derivatives as selected event timestamps. This paper argues that the continuous nature of the polynomial representation limits the ability of E-CIR to generate sharp videos. On the other hand, the proposed spiking representation resembles the same biological principle that is followed by event cameras determining how neurons communicate with each other in living organisms, allowing DeblurSR to represent sharp discontinuous edges and enjoy strong interpretability.

Method

Event Camera Principles

On a pixel grid with resolution $h \times w$, let $L_{xy}(t)$ be the intensity of pixel $(x, y)$ at time $t$. In the natural logarithmic space, consider the amount of intensity change from the previous timestamp $t'$ to the current timestamp $t$:

$$\Delta \ln[L_{xy}(t)] = \ln[L_{xy}(t)] - \ln[L_{xy}(t')]$$

At time $t$, an event, $(x, y, t, p)$, indicates that for pixel $(x, y)$, the instantaneous intensity change exceeds the event generation threshold:

$$\Delta \ln[L_{xy}(t)] \geq c^+, \text{ if } p = 1$$

$$\Delta \ln[L_{xy}(t)] \leq c^-, \text{ if } p = -1$$

where $p = \pm 1$ is the polarity of the event; $c^+$ and $c^-$ are event generation thresholds corresponding to positive events (intensity increments) and negative events (intensity decrements), respectively.

Modern event cameras simultaneously capture a stream of high-speed events and another stream of low-speed conventional frames. Let $B$ be the $h \times w$ conventional frame captured during an exposure interval $[-\frac{T}{2}, \frac{T}{2}]$ with length $T$. Mathematically, the conventional frame is the temporal average of the true physical intensities:

$$B_{xy} = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} L_{xy}(t)dt \quad (3)$$

During the exposure interval, relative motion between the camera and environment leads to undesirable blurriness in the conventional frame capture $B$, which is a visual artifact commonly observed by unprofessional photographers.

Problem Description

As illustrated by Figure 1, the input to the event-based motion deblurring problem includes a blurry image $B = \{B_{xy}\}$ and all the events $E = \{(x, y, t, p) | -\frac{T}{2} \leq t \leq \frac{T}{2}\}$ occurred during the exposure interval. The output of the problem is a sharp video with a temporal range of $[-\frac{T}{2}, \frac{T}{2}]$.

Prediction Algorithm

The Spiking Representation. For each pixel $(x, y)$, we propose to approximate its latent intensity $L_{xy}(t)$ as a parametric mapping from the temporal space $[-\frac{T}{2}, \frac{T}{2}]$ to the normalized intensity space $[0, 1]$. This contrasts with the conventional video representation, where $L_{xy}(t)$ is characterized by discrete samples uniformly distributed across the exposure interval, and enjoys the advantage of having an infinitely high frame rate. Given a blurry image and its associated events, our algorithm learns to predict the coefficients of latent intensity parametric mappings for all the pixels. As shown in Figure 2, the proposed spiking representation uses disconnected line segments to approximate the per-pixel intensity mappings. Within each segment, the coefficients include the slope and the intercept.

Automatic Keypoint Selection Scheme. As illustrated in Figure 2, a piecewise linear function with $n$ pieces has $n + 1$ endpoints. In this paper, we use the term keypoints to refer to the timestamps of these endpoints when the spikes happen. For each pixel $(x, y)$, the sharp intensity $L_{xy}(t)$ is parameterized by $n$ slopes, $n$ intercepts, and $n + 1$ keypoints:

$$L_{xy}(t) \approx \begin{cases} 
  m_{xy_1}x + b_{xy_1}, & t_1 \leq t < t_{xy_1} \\
  m_{xy_2}x + b_{xy_2}, & t_{xy_1} \leq t < t_{xy_2} \\
  \vdots & \\
  m_{xy_n}x + b_{xy_n}, & t_{xy_n} \leq t \leq t_{xy_{n+1}}
\end{cases} \quad (4)$$

Semantically, a keypoint represents a critical moment when the pixel’s intensity changes significantly, which has a natural correlation to the event generation model described in Equation (2). However, the raw event data is highly irregular. During the exposure interval, different pixels have vastly different numbers of events, presenting a substantial challenge for efficient computation in parallel. Events are also known to be noisy (Pan et al. 2019; Wang et al. 2020; Zhang and Yu 2022). It is, therefore, inappropriate to directly extract the raw event timestamps as keypoints. To effectively exploit the correlation between events and keypoints while allowing efficient parallel computation, we propose the automatic keypoint selection module that is robust against noise. Given the blurry image and its associated events, we employ a neural network to predict a set of $n$ segment widths.
Neuromorphic Events

Convolutional Neural Network

UNet

Conv

Coordinate Embedding

Image Embedding

Per-Pixel Feature

(x, y)

Coordinate Embedding

Per-Pixel Feature

Fully-Connected Layers

Spiking Parameters

m_{xyi}; slope
b_{xyi}; intercept
t_{xyi}; keypoint

Figure 3: The Overall Pipeline. Given a blurry image and its associated events in the exposure interval, we apply a Convolutional Neural Network (CNN) to extract an image embedding with the same spatial resolution as the input. For each pixel \((x, y)\), we fuse the image embedding with the coordinate embedding using the addition operation. A group of fully-connected layers take the resulting per-pixel feature vector as input and regress the spiking parameters for each pixel as output. At time \(t_r\), we assemble a spatially varying kernel from the predicted spiking parameters. The convolution of this kernel with the input blurry image gives the output sharp frame at time \(t_r\). By changing the timestamps, the spiking representation allows DeblurSR to render a sharp video with an arbitrarily high frame rate.

for each pixel: \(\{w_{xy1}, w_{xy2}, \ldots, w_{xyN}\}\). Through activation functions, we normalize the sum of these widths to the exposure length, \(T\). The keypoints are \(t_{xy1} = -\frac{T}{2}\) and:

\[ t_{xyi} = t_{xyi-1} + w_{xyi-1} \quad i = 2, 3, \ldots, n + 1 \] (5)

An alternative to the automatic keypoint selection scheme is to use a heuristic-based algorithm to choose the keypoints from all event timestamps. This approach is similar to how E-CIR decides the Lagrange bases for the polynomial functions.

**Integral Normalization Constant.** The basic version of the spiking representation discussed above may not satisfy the physical model in Equation (3) if the coefficients are predicted by a neural network. To fully utilize the information in the blurry input image, we amend the spiking representation in Equation (4) with a normalization constant \(c_{xy}\):

\[ L_{xy}(t) \approx c_{xy} + \begin{cases} m_{xy1} t + b_{xy1} & t_{xy1} \leq t < t_{xy2} \\ m_{xy2} t + b_{xy2} & t_{xy2} \leq t < t_{xy3} \\ \vdots \\ m_{xyN} t + b_{xyN} & t_{xyN} \leq t \leq t_{xyN+1} \end{cases} \] (6)

where, given \(m_{xyi}', b_{xyi}', \text{ and } B_{xy}\), the value of \(c_{xy}\) can be solved analytically from Equation (3) by calculating the indefinite integral of Equation (6):

\[ c_{xy} = -B_{xy} + \frac{1}{2T} \sum_{i=1}^{n} m_{xyi} (t_{xyi+1}^2 - t_{xyi}^2) + \] (7)

\[ \frac{1}{T} \sum_{i=1}^{n} b_i (t_{xyi+1} - t_{xyi}) \] (8)

**Kernel-Based Spatial Modeling.** The sharpness of edges largely determines the quality of motion deblurring. While the above discussion addresses temporal intensity changes, it fails to incorporate spatial brightness variations, which is also key to edge sharpness. To address this issue, we predict \(k^2\) different sets of spiking parameters for each pixel and use them to assemble a \(k \times k\) convolutional kernel \(K_{xy}(t)\). After the spiking representation is further enhanced with spatial kernels, the sharp intensity is given as:

\[ L_{xy}(t) \approx c_{xy} + K_{xy}(t) \odot \mathcal{N}(B_{xy}) \] (9)

where \(\odot\) stands for the inner product operator between two flattened vectors, and \(\mathcal{N}(B_{xy})\) is a \(k \times k\) neighborhood for pixel \((x, y)\) in the input blurry image. We refer interested readers to our open-source GitHub repository for implementation details.

**Prediction Pipeline.** We present the overall motion deblurring pipeline in Figure 3. We first voxelize the irregular event
input into an $m \times h \times w$ histogram tensor (Zhu et al. 2019), where $m$ is the number of histogram bins. We then concatenate this histogram tensor with the blurry image, creating an input tensor whose dimensions are $(m+1) \times h \times w$. A UNet (Ronneberger, Fischer, and Brox 2015)-based convolutional neural network takes this concatenated tensor as input and extracts a $d \times h \times w$ dimensional image embedding. Next, we utilize a linear layer to predict another $d$-dimensional coordinate embedding from the 2D coordinates of each pixel. We fuse the pixel-wise image and coordinate embeddings by the addition operation and employ fully-connected layers to regress the spiking parameters, including $k^2$ slopes, $k^2$ intercepts, and $n$ widths between $n+1$ keypoints for each pixel, where $k$ is the spatial kernel size. The fully connected layers have a total output size of $2k^2+n$. Finally, to render a sharp video with a higher resolution $h\times w$, we apply temporal averaging to generate the midpoint of two regular pixels. This allows natural support to video super resolution. Formally, given a blurry image and events defined at $n$ keypoints for each pixel, we assemble the spatial kernels at rendering timestamps $\{t_r_1, \ldots, t_r_N\}$ and convolve the kernels with the blurry image. The entire pipeline is trained end-to-end using the L1 loss on reconstructed sharp frames.

**Extension to Super Resolution.** Thanks to coordinate embedding, DeblurSR can predict the spiking parameters for pixels with non-integer coordinates in between two regular pixels. During testing, the coordinates take non-integer values such as $(1.5, 7)$, representing the midpoint of two regular pixels. This allows natural support to video super resolution. Formally, given a blurry image and events defined on an $h \times w$ grid, the super-resolution problem aims at reconstructing a sharp video with a higher resolution $h' \times w'$. In the evaluation section, we show that DeblurSR achieves state-of-the-art super-resolution performance even without any high-resolution training supervision.

**Evaluation**

**Experimental Setup**

We conduct experimental evaluations on two benchmark datasets. The REalistic and Dynamic Scenes (REDS) (Nah et al. 2019) dataset is a popular dataset used to evaluate deblurring approaches. The original REDS dataset contains sharp videos with various real-world contents released under the CC BY 4.0 license. Following Song et al. (2022), we use the ESIM simulator (Rebecq, Gehrig, and Scaramuzza 2018) to synthesize events and blurry images. We then employ the official training and validation splits to train and test our model, respectively. Notably, this dataset is also referred to as the GroPro dataset by some authors (Wang et al. 2020), although a much smaller dataset popular in image-to-image deblurring approaches happens to share the same name (Nah, Kim, and Lee 2017).

The High Quality Frames (HQF) (Stoffregen et al. 2020) dataset is another benchmark recently developed to evaluate event-based vision algorithms. The dataset is available for public download, but the licensing details are unclear. The HQF dataset contains both sharp videos and the associated event captures. We apply temporal averaging to generate blurry images from the sharp video. Following Zhang et al. (2022), we use five clips for testing and nine clips for training.

We compare DeblurSR to all the event-based image-to-video motion deblurring approaches with a complete (data preparation, training, and testing) open-source implementation known to us at the time of paper writing. We additionally evaluate eSL-Net (Wang et al. 2020) by creating customized training scripts for the incomplete code released on GitHub.

**Training Details**

We implement DeblurSR under PyTorch (Paszke et al. 2019) and utilize ADAM (Kingma and Ba 2014) to train the network for 50 epochs. We set the initial learning rate to 0.0001 and reduce the learning rate by half after 20 and 40 epochs, respectively. The number of line segments in the spiking representation is $n=10$. The dimension of spatial kernels is $k=3$. The number of histogram bins is $m=26$. The size of image and coordinate embeddings is $d=256$. More details of our experiments are available in the open-source GitHub repository.

**Motion Deblurring**

**Baseline Comparison.** Table 1 presents the quantitative evaluation on the REDS and HQF datasets. We use three image quality metrics to compare DeblurSR with different baseline approaches: the Mean Squared Error (MSE), the Peak Signal-to-Noise Ratio (PSNR), and the Structural Similarity Index Measure (SSIM).

Our method demonstrates an impressive ability in motion deblurring. Specifically, on the REDS dataset, DeblurSR improves the current best-performing method by 12.3% in MSE, 4.7% in PSNR, and 4.9% in SSIM. On HQF, DeblurSR outperforms the state-of-the-art approach by 22.2% in MSE, 10.1% in PSNR, and 14.0% in SSIM.

Qualitatively, as shown in Figure 4 and Figure 5, DeblurSR generates smooth and sharp frames. In particular, we point out that our results are sharper than the EDI reconstruction, which assumes all events correspond to the same amount of intensity change. Meanwhile, our reconstructed frames are significantly less noisy than the eSL-Net reconstruction, which overly emphasizes texture details. Compared with E-CIR, DeblurSR offers more realistic details around the thin edges. The difference between E-CIR and DeblurSR is sometimes subtle and hard to notice from static images.

**Efficiency.** The proposed DeblurSR is computationally efficient. On REDS, it takes 100 hours to train E-CIR for 50 epochs using three Tesla V100 GPUs. By contrast, DeblurSR only requires 72 hours and two of the same GPUs under the identical training setting, representing a 28.0% reduction in training time and a 33.3% reduction in resource demand. The efficiency comes from the simplicity of our spiking representation in contrast with the high-order polynomial parameterization in E-CIR, which allows faster operations like computing the derivative and integral.

**Ablation Study.** Table 2 summarizes the quality of motion deblurring using different variants of the model. The comparison between the first and second rows shows that the proposed automatic keypoint selection module improves the heuristic-based keypoint selection algorithm in E-CIR.
Table 1: On both the REDS and the HQF datasets, DeblurSR outperforms baseline approaches that represent videos by frames and polynomials in terms of the quantitative motion deblurring quality.
Table 2: We use ablation studies to demonstrate the strengths of the automatic keypoint selection scheme (AK), the integral normalization constant (IN), as well as kernel-based spatial modeling (KS).

<table>
<thead>
<tr>
<th>Modules</th>
<th>MSE ↓</th>
<th>REDS</th>
<th>HQF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AK IN KS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>✗ ✗ ✗</td>
<td>0.107 ± 0.001</td>
<td>0.236 ± 0.002</td>
</tr>
<tr>
<td>2</td>
<td>✓ ✗ ✗</td>
<td>0.103 ± 0.001</td>
<td>0.229 ± 0.001</td>
</tr>
<tr>
<td>3</td>
<td>✓ ✓ ✗</td>
<td>0.102 ± 0.001</td>
<td>0.163 ± 0.001</td>
</tr>
<tr>
<td>4</td>
<td>✓ ✓ ✓</td>
<td>0.100 ± 0.001</td>
<td>0.161 ± 0.001</td>
</tr>
</tbody>
</table>

Table 3: We extend DeblurSR to super-resolution and evaluate the performance on the REDS dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSE ↓</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDI (+bicubic)</td>
<td>0.196</td>
<td>20.789</td>
<td>0.550</td>
</tr>
<tr>
<td>eSL-Net</td>
<td>0.228</td>
<td>19.507</td>
<td>0.483</td>
</tr>
<tr>
<td>E-CIR (+bicubic)</td>
<td>0.142</td>
<td>23.530</td>
<td>0.622</td>
</tr>
<tr>
<td>Ours (LR supervision)</td>
<td>0.140</td>
<td>23.642</td>
<td>0.634</td>
</tr>
<tr>
<td>Ours (HR supervision)</td>
<td><strong>0.131</strong></td>
<td><strong>24.272</strong></td>
<td><strong>0.664</strong></td>
</tr>
</tbody>
</table>

Super Resolution

As illustrated in Figure 3, DeblurSR naturally supports super resolution because the pixel coordinates \((x, y)\) can take non-integer values such as \((1.5, 7)\), representing the midpoint of two regular pixels. This section compares DeblurSR with three different baseline approaches on the REDS dataset. We provide the blurry image and the events in low resolution \((180 \times 240)\) and evaluate the results in high resolution \((720 \times 960)\). Among the three baseline approaches, we note that only eSL-Net uses ground-truth high-resolution frames in the training objective. Neither EDI nor E-CIR provides intrinsic support to super-resolution. We obtain high-resolution outputs from EDI and E-CIR through bicubic interpolation. From Table 3, we observe that DeblurSR achieves state-of-the-art performance with and without high-resolution supervision. In the absence of high-resolution supervision, DeblurSR improves E-CIR by 1.4% in MSE. With high-resolution supervision, the relative improvement becomes 7.8%. Figure 6 further confirms our advantage through qualitative visualizations.

Conclusion

In this paper, we introduce DeblurSR, a novel event-based motion deblurring approach based on the spiking representation. DeblurSR builds upon the same biological principles followed by the event camera design. Experiments show that DeblurSR outperforms state-of-the-art approaches in deblurring quality and can be easily extended to video super-resolution. In the future, we plan to modify DeblurSR and allow different pixels to have a different number of parametric segments. This requires a non-trivial redesign of the prediction network to handle the heterogeneity. Another possible direction is to construct a general-purpose deep network with layers of spiking neurons.

Acknowledgments

This research was supported in part by a grant from the NIH DK129979, in part from the Peter O’Donnell Foundation, the Michael J Fox Foundation, Jim Holland-Backcountry Foundation, and in part by a grant from the Army Research Office accomplished under Cooperative Agreement Number W911NF-19-2-0333. Additionally, we acknowledge the support from NSF Career IIS-2047677, NSF HDR-1934932, and NSF CCF-2019844.


