Neuromorphic Event Signal-Driven Network for Video De-Raining

Chengjie Ge, Xueyang Fu*, Peng He, Kunyu Wang, Chengzhi Cao, Zheng-Jun Zha

University of Science and Technology of China, China
cjge@mail.ustc.edu.cn, xyfu@ustc.edu.cn, {hp0618, kunyuwang, chengzhicao}@mail.ustc.edu.cn, zhazj@ustc.edu.cn

Abstract

Convolutional neural networks-based video de-raining methods commonly rely on dense intensity frames captured by CMOS sensors. However, the limited temporal resolution of these sensors hinders the capture of dynamic rainfall information, limiting further improvement in de-raining performance. This study aims to overcome this issue by incorporating the neuromorphic event signal into the video de-raining to enhance the dynamic information perception. Specifically, we first utilize the dynamic information from the event signal as prior knowledge, and integrate it into existing de-raining objectives to better constrain the solution space. We then design an optimization algorithm to solve the objective, and construct a de-raining network with CNNs as the backbone architecture using a modular strategy to mimic the optimization process. To further explore the temporal correlation of the event signal, we incorporate a spiking self-attention module into our network. By leveraging the low latency and high temporal resolution of the event signal, along with the spatial and temporal representation capabilities of convolutional and spiking neural networks, our model captures more accurate dynamic information and significantly improves de-raining performance. For instance, our network achieves a 1.24dB improvement on the *SynHeavy25* dataset compared to the previous state-of-the-art method, while utilizing only 39% of the parameters.

Introduction

Videos captured by traditional vision sensors often suffer degradations in rainy weather. Rain streaks can change the intensity of video frames, obstructing and obscuring the foreground and background information (Li et al. 2022b). In recent years, convolutional neural networks (CNNs) have shown remarkable results in the field of video frames de-raining (VFD). However, most existing CNN-based methods heavily rely on dense intensity frames captured by CMOS sensors, which have limited temporal resolution. The limitation of CMOS sensors poses a challenge in effectively capturing dynamic rainfall information, which hinders further improvements in de-raining performance. This challenge is typically manifested as insufficient de-raining, leaving behind rain streak residue, or excessive de-raining, resulting in texture loss. To address the existing challenge, this study proposes a novel approach that incorporates the neuromorphic signal generated by the event camera into the video de-raining. The event camera is a bio-inspired vision sensor designed to detect and record spatial-temporal changes for each pixel. Compared to CMOS-based cameras, the event camera offers several advantages (Gallego et al. 2020), including low temporal latency (1μs vs. 1ms), high dynamic range (140dB vs. 60dB), and low power consumption (10mW vs. 10W).

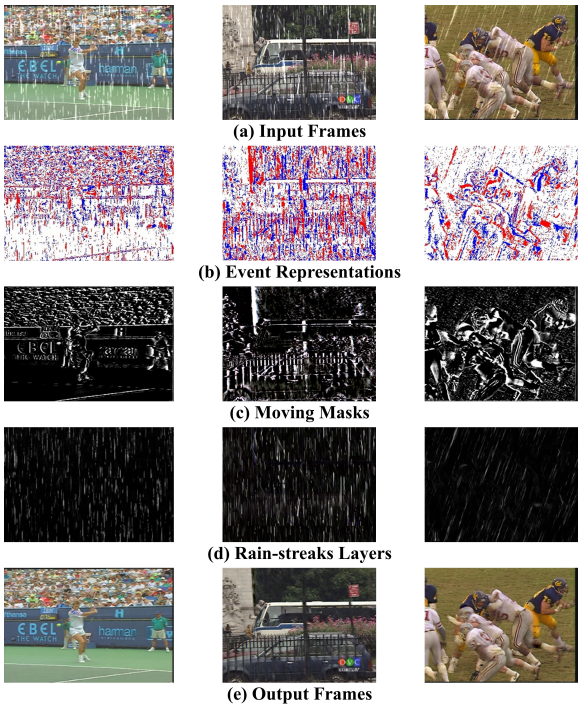


Figure 1: Visualizations of some elements in our de-raining model in successive frames.

Based on the mechanism of the event camera, we can record additional dynamic information through event streams generated by rapid rain streaks and moving objects. These event streams provide us with nearly continuous views of external dynamic information, allowing us to overcome the limitations of dynamic feature perception and

*Corresponding author.

improve the de-raining performance.

The contributions of this paper can be summarized as follows:

- We utilize the dynamic information provided by the event signal as prior knowledge, and integrate it into the existing video de-raining objective. The modified objective includes stationary backgrounds, rain-streak layers, and moving objects. By incorporating the event signal, we can impose a better constraint on the solution space.
- We propose an optimization algorithm to solve the modified objective function, and design a deep unfolding network that mimics the optimization process. This network combines CNNs and spiking neural networks (SNNs), allowing us to seamlessly integrate two modes of signals, namely event streams and video frames, into our deep unfolding network.
- Within the proposed network, we embed a well-designed spiking self-attention module. This module enables our model to selectively focus on relevant spatial-temporal regions, capturing more accurate dynamic information from event streams and leading to better rain removal.

We update parameters from corresponding dense frames and sparse streams to achieve data-driven learning for different priors. Extensive experiments demonstrate the superiority of additional dynamic priors provided by the event streams both quantitatively and visually. For example, our network achieves a 1.24dB improvement on *SynHeavy25* (Yang et al. 2019), a 1.18dB improvement on *SynLight25* (Yang et al. 2019), and a 0.52dB improvement on *NTURain* (Chen et al. 2018) over the previous state-of-the-art method (Yang et al. 2021) with only 39% of the parameters. Additionally, to validate the effectiveness of our method in real-world scenarios, we captured several real rainy scenes using an event camera. By combining the low latency and high temporal resolution of event signals with the spatial-temporal representation capabilities of CNNs and SNNs, our network achieves favorable de-raining performance on these authentic sequences.

Related Works

Video De-Raining

Unlike single-image de-raining (SID) methods, video frame de-raining (VFD) methods leverage the temporal correlation among frames to predict clear backgrounds. Traditional VFD models use prior-based approaches to retrieve the temporal and motion context (Chen et al. 2018; Liu et al. 2018; Mu et al. 2021; Yang, Liu, and Feng 2019), treating rain streaks at the photometric level. Recently, with the advancements in deep learning, data-driven de-raining methods have gained dominance in the VFD field (Yue et al. 2021; Yan et al. 2021, 2022; Yang et al. 2022; Mu et al. 2021). For example, Yang *et al.* propose a Self-Learned de-raining Network (SLDNet+) that explores the temporal correlation, consistency, and rain-related priors among rain frames (Yang et al. 2022). Mu *et al.* present a model-driven triple-level model optimization framework (TMICS) that infers the network architecture using cooperative optimization and an auto-searching mechanism (Mu et al. 2021). Zhang *et al.* propose a light-weight video de-raining network called ESTINet, which utilizes SICM and STIM blocks during the training phase to capture spatial-temporal features and reconstruct video frames. These frames are then further refined using a 3D-DenseNet ESTM (Zhang et al. 2022).

Event-Enhanced Reconstruction

The event camera, a bio-inspired vision sensor, offers several advantages compared to traditional cameras (Gallego et al. 2020). These advantages include low temporal latency, high dynamic range, and low power consumption. Event cameras excel in handling sophisticated scenarios which pose challenges for traditional cameras (Zhang et al. 2021b; Zhu et al. 2019; Rebecq et al. 2019b; Zhang et al. 2020; Weng, Zhang, and Xiong 2021; Sun et al. 2022, 2023; Pan et al. 2019). While intensity frames provide detailed textures, event streams provide rich temporal details, making them complementary sources for video construction. For example, Wang *et al.* propose an interpretable network that performs low-level tasks such as denoising, deblurring, and super-resolution under the guidance of the event stream (Wang et al. 2020a). They also introduce a synthesized dataset for video deblurring. Zhang *et al.* present a deep fine-grained video deblurring pipeline by combining the blurry image with modified event streams at a modified temporal period (Zhang et al. 2021a). Cao *et al.* propose an end-to-end network that utilizes spatial and temporal features from event streams and fuses event features into the intensity frame for video deblurring (Cao et al. 2022).

Building upon these studies, we introduce event streams into the VFD, utilizing them as dynamic priors. In our approach, we construct a deep unfolding network that combines SNNs for processing event streams and CNNs for handling intensity frames.

Methodology

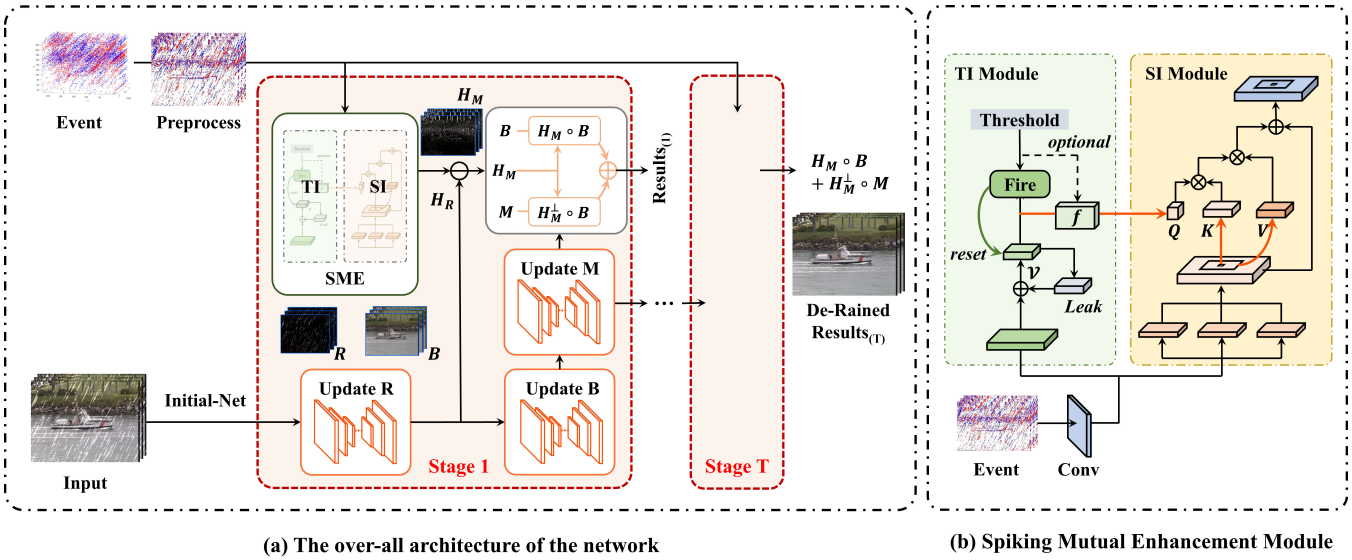
In general, an input video sequence can be expressed as $\mathcal{O} \in \mathbf{R}^{h\times w\times s}$, where h, w, and s stand for the height, width, and the frames number of the video sequence, accordingly. Following the above mentioned analysis, the VFD model can be denoted as:

$$\mathcal{O} = \mathcal{B} + \mathcal{R} + \mathcal{M}, \tag{1}$$

where $\mathcal{B}$, $\mathcal{R}$, and $\mathcal{M} \in \mathbf{R}^{h\times w\times s}$ represent the stationary backgrounds, rain-streaks layers, and moving objects of the video sequence.

Model Formulation

Modeling Rain-Streaks Layers and Background Layers: Inspired by K-SVD (Aharon, Elad, and Bruckstein 2006), the video frame $\mathbf{Y}$ can be decomposed as the multiples of sparse dictionary matrices $\mathcal{D}$ and sparse encoding coefficient matrices $\mathcal{C}$, where $\mathcal{D}$ stores the information in $\mathbf{Y}$, and $\mathcal{C}$ represents how those features were combined. Thus, the model of rain-streaks layers and background layers is formulated as follows to capture the sophisticated features embedded in



(a) The over-all architecture of the network (b) Spiking Mutual Enhancement Module

Figure 2: (a) The over-all architecture of the network. The Initial-Net is a ResBlock to generate initialization value for each elements. The network consists of T stages with four sub-updating modules. At each stage, our network updates the variables in order. (b) The Spiking Mutual Enhancement Module (SME Module), which consists of two parts: the Temporal Interaction Module (TI Module) which includes a Full-precision LIF Module, and the Spatial Interaction Module (SI Module).

corresponding layers:

$$\mathcal{R} = \sum_{k=1}^{K} \mathcal{D}_{\mathcal{R}}^{(k)} \otimes \mathcal{C}_{\mathcal{R}}^{(k)}, \quad \mathcal{B} = \sum_{k=1}^{K} \mathcal{D}_{\mathcal{B}}^{(k)} \otimes \mathcal{C}_{\mathcal{B}}^{(k)}. \tag{2}$$

In Eq. (2), k represents the number of feature layers. $\mathcal{D}_{\mathcal{B}}$ stands for a series of matrices with k feature maps that store the fine texture in the background layer. $\mathcal{C}_{\mathcal{B}}$ stands for k feature layers, and maps the dictionary matrix to the clear background. The same interpretation applies to $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{C}_{\mathcal{R}}$.

Modeling Moving Objects With Event Streams: For traditional frame-based de-raining methods, the moving objects layer is hard to predict because the variation among frames is uncertain. Previous video de-raining methods apply Markov Random Fields (MRF) as a binary mask $\mathcal{H}$ to outline the moving objects (Li et al. 2018, 2021). However, the extensive and rapid rain streaks will diffuse the gap between the foreground and background, thus resulting in the failure of the graph-cut algorithm. To this issue, we reformulate the model of moving masks as the combination of moving rain streaks and moving objects:

$$\mathcal{H}_{\mathcal{O}} = \mathcal{H}_{\mathcal{M}} + \mathcal{H}_{\mathcal{R}}, \tag{3}$$

where $\mathcal{H}_{\mathcal{O}}$, $\mathcal{H}_{\mathcal{M}}$, and $\mathcal{H}_{\mathcal{R}}$ stand for the whole moving masks, moving objects masks and rain masks. As we mentioned before, event cameras are able to capture the difference between frames with ultra high time resolution, which means that after moderate data-processing, the event streams can be treated as the dynamic priors, hence greatly reduce the optimization complexity. To further decouple $\mathcal{H}_{\mathcal{M}}$ from $\mathcal{H}_{\mathcal{O}}$, we introduce the end-to-end learning blocks in the following section. As to the moving objects layers, they satisfy the following equations:

$$\mathcal{H}_{\mathcal{M}} \circ \mathcal{O} = \mathcal{H}_{\mathcal{M}} \circ \mathcal{M}, \tag{4}$$

where $\circ$ stands for the Hadamard product. Noticed that in Eq. (4), previous methods assume that the moving objects layer $\mathcal{M}$ satisfy the smooth property (Li et al. 2021). However, this prior knowledge fails to work when the object moves rapidly. In this case, we add another adaptive penalty item to regularize the $\mathcal{M}$ layers.

Taking all factors into account, the over-all augmented Lagrangian function can be established:

$$\begin{aligned}
\mathcal{L}_{\sigma} = & \|\mathcal{H}_{\mathcal{M}} \circ (\mathcal{O} - \mathcal{M})\|_F^2 + \lambda_m \phi_m(\mathcal{H}_{\mathcal{M}}) \\
& + \lambda_{\mathcal{D}_{\mathcal{R}}} \phi_{\mathcal{D}_{\mathcal{R}}}(\sum_{k=1}^{K} \mathcal{D}_{\mathcal{R}}^{(k)}) + \lambda_{\mathcal{D}_{\mathcal{B}}} \phi_{\mathcal{D}_{\mathcal{B}}}(\sum_{k=1}^{K} \mathcal{D}_{\mathcal{B}}^{(k)}) \\
& + \lambda_{\mathcal{C}_{\mathcal{R}}} \psi_{\mathcal{C}_{\mathcal{R}}}(\sum_{k=1}^{K} \mathcal{C}_{\mathcal{R}}^{(k)}) + \lambda_{\mathcal{C}_{\mathcal{B}}} \psi_{\mathcal{C}_{\mathcal{B}}}(\sum_{k=1}^{K} \mathcal{C}_{\mathcal{B}}^{(k)}) \\
& + \lambda_{\mathcal{M}} \Gamma_{\mathcal{M}}(\mathcal{M}) + \frac{\sigma_{\mathcal{B}}}{2} \|\sum_{k=1}^{K} \mathcal{D}_{\mathcal{B}}^{(k)} \otimes \mathcal{C}_{\mathcal{B}}^{(k)} - \mathcal{B}\|_F^2 \\
& + \frac{\sigma_{\mathcal{R}}}{2} \|\sum_{k=1}^{K} \mathcal{D}_{\mathcal{R}}^{(k)} \otimes \mathcal{C}_{\mathcal{R}}^{(k)} - \mathcal{R}\|_F^2,
\end{aligned} \tag{5}$$

where σ denotes the penalty item, $\{\phi, \psi, \Gamma\}$ are the deep priors of corresponding elements, and λ are the hyper-parameters. For the sake of expression simplicity, we omit the superscript and summation symbols in the following description.

Deep Unfolding Optimization Algorithm

It should be noticed that the over-all function is a non-convex problem, so we apply a multi-step solution to this

problem. The network structure is shown in Figure 2(a).

(I). Updating $\mathcal{R}$: The sub-problem in terms of $\mathcal{R}$ is:

$$\begin{aligned}\min_{\mathcal{D}_\mathcal{R},\mathcal{C}_\mathcal{R}} \frac{\sigma_\mathcal{R}}{2}\|\mathcal{D}_\mathcal{R}\otimes\mathcal{C}_\mathcal{R}-\mathcal{R}\|_F^2 \\ +\lambda_{\mathcal{D}_\mathcal{R}}\phi_{\mathcal{D}_\mathcal{R}}(\mathcal{D}_\mathcal{R})+\lambda_{\mathcal{C}_\mathcal{R}}\psi_{\mathcal{C}_\mathcal{R}}(\mathcal{C}_\mathcal{R}).\end{aligned} \tag{6}$$

Eq. (6) is a bi-criterion optimization problem, so it can be solved by updating $\mathcal{D}_\mathcal{R}$ and $\mathcal{C}_\mathcal{R}$ accordingly (Zhang, Gool, and Timofte 2020).

Updating $\mathcal{D}_\mathcal{R}$: The optimization problem of $\mathcal{D}_\mathcal{R}$ is:

$$\min_{\mathcal{D}_\mathcal{R},\mathcal{C}_\mathcal{R}} \frac{\sigma_\mathcal{R}}{2}\|\mathcal{D}_\mathcal{R}\otimes\mathcal{C}_\mathcal{R}-\mathcal{R}\|_F^2+\lambda_{\mathcal{D}_\mathcal{R}}\phi_{\mathcal{D}_\mathcal{R}}(\mathcal{D}_\mathcal{R}). \tag{7}$$

$\phi_{\mathcal{D}_\mathcal{R}}(\mathcal{D}_\mathcal{R})$ is the penalty item of Eq. (7). We introduce the variable $\widehat{\mathcal{D}}_\mathcal{R}$ to decouple while simplify calculations. The modified Lagrangian function can be expressed as:

$$\begin{aligned}\min_{\mathcal{D}_\mathcal{R},\mathcal{C}_\mathcal{R}} \frac{\sigma_\mathcal{R}}{2}\|\widehat{\mathcal{D}}_\mathcal{R}\otimes\mathcal{C}_\mathcal{R}-\mathcal{R}\|_F^2 \\ +\lambda_{\mathcal{D}_\mathcal{R}}\phi_{\mathcal{D}_\mathcal{R}}(\mathcal{D}_\mathcal{R})+\frac{\mu_{\mathcal{D}_\mathcal{R}}}{2}\|\widehat{\mathcal{D}}_\mathcal{R}-\mathcal{D}_\mathcal{R}\|_F^2,\end{aligned} \tag{8}$$

where $\mu_{\mathcal{D}_\mathcal{R}}$ are the hyperparameters. Considering of simplicity and fast convergence, we apply HQS algorithm (He et al. 2013) to unfold Eq. (8) iteratively.

$$\begin{aligned}\widehat{\mathcal{D}}_{\mathcal{R}_{(t+1)}} = \min_{\widehat{\mathcal{D}}_\mathcal{R}} \frac{\sigma_\mathcal{R}}{2}\|\widehat{\mathcal{D}}_{\mathcal{R}_{(t)}}\otimes\mathcal{C}_\mathcal{R}-\mathcal{R}\|_F^2 \\ +\frac{\mu_{\mathcal{D}_\mathcal{R}}}{2}\|\widehat{\mathcal{D}}_{\mathcal{R}_{(t)}}-\mathcal{D}_{\mathcal{R}_{(t)}}\|_F^2,\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{D}_{\mathcal{R}_{(t+1)}} = \min_{\mathcal{D}_\mathcal{R}} \frac{\mu_{\mathcal{D}_\mathcal{R}}}{2}\|\widehat{\mathcal{D}}_{\mathcal{R}_{(t+1)}}-\mathcal{D}_{\mathcal{R}_{(t)}}\|_F^2 \\ +\lambda_{\mathcal{D}_\mathcal{R}}\phi_{\mathcal{D}_\mathcal{R}}(\mathcal{D}_{\mathcal{R}_{(t)}}).\end{aligned} \tag{10}$$

Notably, the closed form solution of $\widehat{\mathcal{D}}_\mathcal{R}$ can be obtained by taking the partial derivative of Eq. (9) with respect to $\widehat{\mathcal{D}}_\mathcal{R}$ and deriving the zero point of the equation:

$$\mathbf{F^{-1}}\{(\boldsymbol{\sigma_\mathbf{R}}\mathbf{C_R^H}\mathbf{C_R}+\boldsymbol{\mu_{\mathbf{D_R}}}\mathbf{I})^{-1}(\boldsymbol{\sigma_\mathbf{R}}\mathbf{C_R^H}\mathbf{R}+\boldsymbol{\mu_{\mathbf{D_R}}}\mathbf{D_R(t)})\}, \tag{11}$$

where $\mathbf{F}$ is the Fast Fourier Transform (FFT); $\mathbf{F^{-1}}$ represents the inverse Fast Fourier Transform (IFFT); $\mathbf{C_R} = \mathbf{F}(\mathcal{C}_\mathcal{R})$, $\mathbf{C_R^H}$ denotes the complex conjugate of $\mathbf{C_R}$.

It is hard to find a closed-form equation for Eq. (10) because we do not assume $\phi_{\mathcal{D}_\mathcal{R}}$ as specific prior knowledge, thus we apply estimation neural networks (EstNet) to estimate prior knowledge from the data by itself. In this way, each dictionary is derived from the sparse encoding metrics, which enables our framework to generate dictionaries according to the input images. The visual results will be shown in the **appendix**.

Updating $\mathcal{C}_\mathcal{R}$:

$$\min_{\mathcal{D}_\mathcal{R},\mathcal{C}_\mathcal{R}} \frac{\sigma_\mathcal{R}}{2}\|\mathcal{D}_\mathcal{R}\otimes\mathcal{C}_\mathcal{R}-\mathcal{R}\|_F^2+\lambda_{\mathcal{C}_\mathcal{R}}\psi_{\mathcal{C}_\mathcal{R}}(\mathcal{C}_\mathcal{R}). \tag{12}$$

Following Eq. (12), we can update $\mathcal{C}_\mathcal{R}$ by introducing an auxiliary variable $\widehat{\mathcal{C}}_\mathcal{R}$ to decouple the original variable and update the equation in a similar two-step manner.

(II). Updating $\mathcal{B}$: The sub-problem in terms of $\mathcal{B}$ is:

$$\begin{aligned}\min_{\mathcal{D}_\mathcal{B},\mathcal{C}_\mathcal{B}} \frac{\sigma_\mathcal{B}}{2}\|\sum_{k=1}^{K}\mathcal{D}_\mathcal{B}\otimes\mathcal{C}_\mathcal{B}-\mathcal{B}\|_F^2 \\ +\lambda_{\mathcal{D}_\mathcal{B}}\phi_{\mathcal{D}_\mathcal{B}}(\mathcal{D}_\mathcal{B})+\lambda_{\mathcal{C}_\mathcal{B}}\psi_{\mathcal{C}_\mathcal{B}}(\mathcal{C}_\mathcal{B}),\end{aligned} \tag{13}$$

which can be easily solved following the same process of **Updating** $\mathcal{R}$. More details are shown in the **appendix**.

(III). Updating $\mathcal{H}_\mathcal{M}$: The sub-problem in terms of $\mathcal{H}_\mathcal{M}$ is:

$$\min_{\mathcal{H}_\mathcal{M}}\|\mathcal{H}_\mathcal{M}\circ\mathcal{O}-\mathcal{H}_\mathcal{M}\circ\mathcal{M}\|_F^2. \tag{14}$$

We first pre-process the event stream, and then use a spiking attention module, which would be discussed in the following section, to extract the features in event streams as the initialized $\mathcal{H}_\mathcal{O}$.

Eq. (14) is equivalent to solve this equation:

$$\begin{aligned}\min_{\mathcal{H}_\mathcal{R}}\|\mathcal{H}_\mathcal{R}\circ\mathcal{R}-\mathcal{R}\|_F^2, \\ \mathcal{H}_\mathcal{M}=\mathcal{H}_\mathcal{O}-\mathcal{H}_\mathcal{R}.\end{aligned} \tag{15}$$

(IV). Updating $\mathcal{M}$: The sub-problem in terms of $\mathcal{M}$ is:

$$\min_{\mathcal{M}}\|\mathcal{H}_\mathcal{M}\circ(\mathcal{O}-\mathcal{M})\|_F^2+\lambda_\mathcal{M}\Gamma_\mathcal{M}(\mathcal{M}). \tag{16}$$

Following the ADMM algorithm (Boyd et al. 2011), Eq. (16) can be solved in a three-step manner. The augmented Lagrangian of Eq. (16) is:

$$\begin{aligned}\mathcal{L}(\mathcal{M};Z;\Theta)=\|\mathcal{H}_\mathcal{M}\circ(\mathcal{O}-\mathcal{Z})\|_F^2+\lambda_\mathcal{M}\Gamma_\mathcal{M}(\mathcal{M}) \\ +\gamma\|\mathcal{M}-(Z-\Theta)\|_F^2.\end{aligned} \tag{17}$$

Noticed that we introduce the variable Θ in Eq. (17) to hasten the convergence. The optimization function can be written as ($\mathcal{Z}$ is a auxiliary variable):

Updating $\mathcal{Z}$:

$$\begin{aligned}\mathcal{Z}_{(t+1)}=\min_{\mathcal{Z}}\|\mathcal{H}_\mathcal{M}\circ(\mathcal{O}-\mathcal{Z}_{(t)})\|_F^2 \\ +\gamma\|\mathcal{M}_{(t+1)}-(\mathcal{Z}_{(t)})-\Theta_{(t)})\|_F^2.\end{aligned} \tag{18}$$

Eq. (18) has a closed-form equation:

$$(\mathcal{H}_\mathcal{M}^\mathcal{H}\mathcal{H}_\mathcal{M}+\boldsymbol{\gamma}\mathcal{I})^{-1}(\boldsymbol{\gamma}\mathcal{M}_{(t+1)}+\boldsymbol{\gamma}\Theta_{(t)}+\mathcal{H}_\mathcal{M}^\mathcal{H}\mathcal{H}_\mathcal{M}\mathcal{O}). \tag{19}$$

Updating $\mathcal{M}$:

$$\begin{aligned}\mathcal{M}_{(t+1)}=\min_{\mathcal{M}}\lambda_\mathcal{M}\Gamma_\mathcal{M}(\mathcal{M}_{(t)}) \\ +\gamma\|\mathcal{M}_{(t)}-(\mathcal{Z}_{(t)}-\Theta_{(t)})\|_F^2.\end{aligned} \tag{20}$$

Eq. (20) can be approximated by a EstNet.

Updating Θ:

$$\Theta_{(t+1)}=\Theta_{(t)}+\rho(\mathcal{M}_{(t+1)}-\mathcal{Z}_{(t+1)}). \tag{21}$$

Eq. (21) is equivalent to a dual ascent step that can be approximated by a proximal operator module (Wang et al. 2023).

Final Results: The final results of the whole network is expressed as:

$$\textbf{De-rained results}=\mathcal{H}_\mathcal{M}\circ\mathcal{M}+(\mathcal{I}-\mathcal{H}_\mathcal{M})\circ\mathcal{B}. \tag{22}$$

Network Design

The main challenge of achieving deep unfolding in the algorithm is how to solve equations with deep priors, such as Eq. (10), and extract the necessary motion priors from event stream data, as described in Eq. (14). In this study, we follow the Figure 2 to build our network and solve the optimization function iteratively. Specifically, we select a ResBlock as the approximation operator for solving the dictionary problem in Eq. (10), and a U-Net structure to estimate sparse coefficients like previous methods (Zhu et al. 2022; Wang et al. 2023). To extract motion priors from the event stream, we utilize a novel Spiking Mutual Enhancement (SME) module to extract features in the temporal and spatial domains as denoted in Figure 2(b), and fused them to generate the initial $\mathcal{H}_{\mathcal{O}}$. Subsequently, we can separately solve the distribution for each module in the network, and implement the following procedures for the entire network.

$$R\text{-}Net \begin{cases} \widehat{\mathcal{D}}_{\mathcal{R}_{(t+1)}} = \mathbf{F^{-1}}\{(\boldsymbol{\sigma}_{\mathbf{R}}\mathbf{C}_{\mathbf{R}}^{\mathbf{H}}\mathbf{C}_{\mathbf{R}} + \boldsymbol{\mu}_{\mathbf{D}_{\mathbf{R}}}\mathbf{I})^{-1} \\ \qquad\qquad (\boldsymbol{\sigma}_{\mathbf{R}}\mathbf{C}_{\mathbf{R}}^{\mathbf{H}}\mathbf{R} + \boldsymbol{\mu}_{\mathbf{D}_{\mathbf{R}}}\mathbf{D}_{\mathbf{R}}(\mathbf{t}))\}, \\ \mathcal{D}_{\mathcal{R}_{(t+1)}} = EstNet(\widehat{\mathcal{D}}_{\mathcal{R}_{(t+1)}}, \lambda_{\mathcal{D}_{\mathcal{R}}}). \end{cases} \tag{23}$$

$$H\text{-}Net \begin{cases} \mathcal{H}_{\mathcal{O}} = SME(Event), \\ \mathcal{H}_{\mathcal{R}} = \min\limits_{\mathcal{H}_{\mathcal{R}}} \|\mathcal{H}_{\mathcal{R}} \circ \mathcal{R} - \mathcal{R}\|_F^2, \\ \mathcal{H}_{\mathcal{M}} = \mathcal{H}_{\mathcal{O}} - \mathcal{H}_{\mathcal{R}}. \end{cases} \tag{24}$$

$$B\text{-}Net \begin{cases} \widehat{\mathcal{D}}_{\mathcal{B}_{(t+1)}} = \mathbf{F^{-1}}\{(\boldsymbol{\sigma}_{\mathbf{B}}\mathbf{C}_{\mathbf{B}}^{\mathbf{H}}\mathbf{C}_{\mathbf{B}} + \boldsymbol{\mu}_{\mathbf{D}_{\mathbf{B}}}\mathbf{I})^{-1} \\ \qquad\qquad (\boldsymbol{\sigma}_{\mathbf{B}}\mathbf{C}_{\mathbf{B}}^{\mathbf{H}}\mathbf{B} + \boldsymbol{\mu}_{\mathbf{D}_{\mathbf{B}}}\mathbf{D}_{\mathbf{B}}(\mathbf{t}))\}, \\ \mathcal{D}_{\mathcal{B}_{(t+1)}} = EstNet(\widehat{\mathcal{D}}_{\mathcal{B}_{(t+1)}}, \lambda_{\mathcal{D}_{\mathcal{B}}}). \end{cases} \tag{25}$$

$$M\text{-}Net \begin{cases} \mathcal{Z}_{(t+1)} = (\mathcal{H}_{\mathcal{M}}^{\mathcal{H}}\mathcal{H}_{\mathcal{M}} + \boldsymbol{\gamma}\mathcal{I})^{-1}(\boldsymbol{\gamma}\mathcal{M}_{(t+1)} \\ \qquad\qquad + \boldsymbol{\gamma}\Theta_{(t)} + \mathcal{H}_{\mathcal{M}}^{\mathcal{H}}\mathcal{H}_{\mathcal{M}}\mathcal{O}), \\ \mathcal{M}_{(t+1)} = EstNet(\mathcal{Z}_{(t+1)}, \delta_{\mathcal{M}}), \\ \Theta_{t+1} = \Theta_t + \rho(\mathcal{M}_{(t+1)} - \mathcal{Z}_{(t+1)}). \end{cases} \tag{26}$$

$$\textbf{De-rained results} = \mathcal{H}_{\mathcal{M}} \circ \mathcal{M} + (I - \mathcal{H}_{\mathcal{M}}) \circ \mathcal{B}. \tag{27}$$

Spiking Mutual Enhancement Modules

SNNs are networks inspired by biological systems that are inherently compatible with the asynchronous and sparse nature of event streams (Roy, Jaiswal, and Panda 2019), employing biomimetic spiking neurons as their fundamental computational units. Inspired by the attention mechanism (Vaswani et al. 2017) and the rapid development of SNNs (Kim et al. 2020; Zhang et al. 2018), we introduce the SME Module to capture temporal and spatial features among event streams. Due to the time-redundancy and sparsity, traditional CNN modules are not suitable for processing event streams, so we adopt SNN blocks to handle event streams in the temporal domain instead. Among SNN blocks, the Full-precision LIF (Li et al. 2022a) shows superior performance on many vision tasks. The principle of Full-precision LIF neurons is depicted as:

$$y_{(t+1)}^n = \sum W^T x, \tag{28}$$

$$u_{(t+1)}^n = \tau u_{(t)}^n (1 - o_{(t)}^n) + y_{(t+1)}^n, \tag{29}$$

$$o_{(t+1)}^n = u_{(t+1)}^n > V_{th}, \tag{30}$$

$$r_{(t+1)}^n = \max(u_{(t+1)}^n, 0). \tag{31}$$

As shown in Figure 2(b), the SME Module takes three successive event features as input, which are firstly fed into a LIF module to excavate temporal features, and then fed into the SI Module to further explore the spatial relevance.

Experiments

Implementation Details

Training Setting: All the experiments are implemented on a NVIDIA RTX 3090 based on Pytorch. We adopt the ADAM optimizer (Kingma and Ba 2014) and the batch size of 4 to train our model. The images are cropped into 128×128 patch size with random horizontal flipping. The total training epoch is set to be 1000. The initial learning rate is set to be 10^{-4}, and divided by 2 every 200 epochs.

Loss Function

Our network is regulated in an end-to-end manner. The whole loss function is:

$$\mathcal{L} = -\sum_{i=1}^{T} \frac{i}{T} \cdot SSIM(\mathbf{GT}, \mathbf{Results_{(i)}}), \tag{32}$$

where $\mathbf{Results_{(i)}}$ indicates the **De-rained results** of our model in the i_{th} stage.

Experiment Results

Datasets: In this section, we compare our method with previous methods on four most commonly used benchmark datasets. *SynHeavy25* and *SynLight25* have the same ground truth images. *SynLight25* is proposed with few and scattered rain streaks, while *SynHeavy25* is embedded with extensive and rapid rain streaks (Yang et al. 2019). *NTURain* (Chen et al. 2018) is synthesized under two scenarios. One is captured by a camera with slow movements, while the other is derived from a fast-moving camera. There are several outstanding event camera simulators and methods in existence (Rebecq, Gehrig, and Scaramuzza 2018; Hu, Liu, and Delbruck 2021; Rebecq et al. 2019a). However, we strictly follow (Duan et al. 2021) to obtain event streams to ensure the authenticity of event streams. For real-world raining frames, we choose a Davis346 event camera to capture the rainy sequence and event streams.

Baselines: To verify the effectiveness of our proposed methods, we compare it with previous state-of-the-art methods: deep detail network (DetailNet) (Fu et al. 2017), joint recurrent rain removal and reconstruction (J4RNet) (Liu et al. 2018), superpixel alignment and compensation CNN (SpacCNN) (Chen et al. 2018), joint rain detection and removal (JORDER) (Yang et al. 2019), deep cross-scale fusion

Metric	DataSet	DetailNet	J4RNet	SpacCNN	DDCDNet	DGCNet	TMICS	RFMD	ESTINet	Ours
PSNR	*NTURain*	30.13	32.77	31.74	36.64	37.46	35.07	38.92	37.48	**39.44**
SSIM		0.9220	0.9540	0.9390	0.9702	0.9769	0.9681	0.9764	0.9700	**0.9821**
PSNR	*SynLight25*	25.72	28.19	25.40	34.57	35.53	36.10	36.99	34.57	**38.17**
SSIM		0.8572	0.8806	0.8260	0.9577	0.9624	0.9674	0.9760	0.9631	**0.9813**
PSNR	*SynHeavy25*	16.50	24.51	23.73	28.47	27.76	28.90	32.70	27.72	**33.94**
SSIM		0.5441	0.7492	0.7310	0.8886	0.8598	0.8743	0.9350	0.8239	**0.9416**

Table 1: Quantitative evaluation of different SOTA methods on datasets *NTURain*, *SynLight25* and *SynHeavy25*. Best and second best indexs are marked in bold and underline.

Figure 3: De-rained results on the video frame in dataset *SynHeavy25*.

network (DCSFN) (Wang et al. 2020b), dual graph convolutional network (DGCNet) (Fu et al. 2021), deep dual convolutional dictionary learning network (DDCDNet) (Ge, Fu, and Zha 2022), triple-level model inferred collaborative network (TMICS) (Mu et al. 2021), recurrent multi-frame deraining (RMFD) (Yang et al. 2021), and enhanced spatial-temporal interaction network (ESTINet) (Zhang et al. 2022). Noticed that DetailNet, JORDER, DCSFN, DGCNet and DDCDNet are SID methods, and J4RNet, SpacCNN, RMFD and ESTINet are VFD methods.

Quantitative Evaluation: Table 1 shows the quantitative comparison among different methods. Prior to our work, the RMFD method took the lead on three common video de-raining datasets especially on dataset *SynHeavy25*. Our method further improves the PSNR and SSIM (Wang et al. 2004) metrics while reducing the number of parameters by 61% (11M) in our network framework compared to the RMFD (29M) method. Our method obtain a 1.24dB improvement on dataset *SynHeavy25*, a 1.18dB improvement on dataset *SynLight25*, and a 0.52dB improvement on dataset *NTURain*. These enhancements are attributed to the introduction of event streams, SME Modules and proper modeling of video de-raining. We will explore these modules in more details in **Ablation Studies**.

Qualitative Evaluation: As for qualitative evaluation, Figure 3 and 4 show the visual comparison between our method and other previous methods. Observed that in Figure 3, there are dense and extensive rain streaks. DCSFN, ESTINet and TMICS fail to remove rain streaks. DDCDNet and RMFD lost some of the textures in the background panel. This is due to the fact that complex and dense rain streaks blur the foreground and background of the video frame, resulting in inadequate de-raining. For comparison, our methods well decouple the rain streaks from the original rain frame, with no rain streaks remaining visible to human eyes while preserving more detailed textures. For the real-world video frame, we choose a real-world video frame from our proposed sequence for visual comparison. As shown in Figure 4, in real-world datasets, the demarcation between the rain-streak layer and the background layer is notably more ambiguous, rendering the remaining SID and VFD methods less effective in removing rain streaks. Conversely, our approach, guided by the introduction of event data, distinguishes the rain-streak layer from the background layer more effectively, thereby demonstrating superior rain removal performance.

Network Parameter and Performance Comparison: In order to demonstrate the superiority of our network in terms of parameter and performance, we conduct a visual comparison between the currently popular de-raining methods and our proposed method, as shown in the Figure 5. The vertical axis represents the PSNR achieved by each method on the *SynHeavy25* dataset, while the horizontal axis represents the parameter of each method. The diameter of the circles in the figure represents the time required for the model to infer a 640×480 frame. As shown in Figure 5, when the base channel of our method is set to 16, we have already achieved the highest PSNR value compared with previous methods. Increasing the number of base channels further can improve performance metrics, but it may not be justifiable in terms of the additional storage and computation requirements. Therefore, we determine that a channel number of 24 strikes an appropriate balance between performance and complexity.

Feature Visualization: To demonstrate features extracted by the SME Module more intuitively, we visualize feature layers learned by SME, as shown in Figure 6. Thanks to the accurate dynamic information provided by event streams, the SME Module can accurately capture rain streaks and moving objects in videos, thus further enhancing the rain removal capability. More visualizations of network modules are shown in the **appendix**.

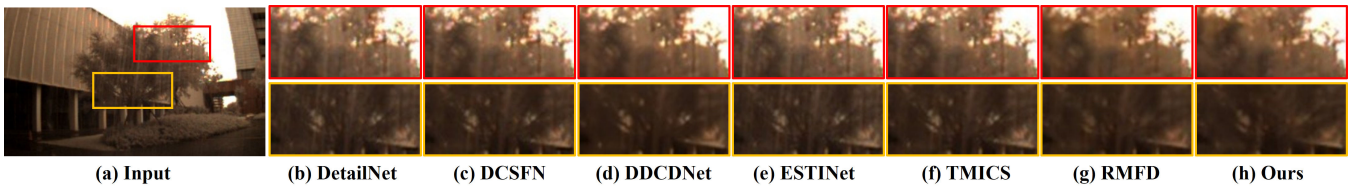


Figure 4: De-rained results on the real-world video frame captured by our event camera.

Methods	DetailNet	JORDER	DDCDNet	TMICS	RFMD	Ours
w/o event	16.50/0.5441	23.28/0.7513	28.47/0.8866	28.90/0.8743	**32.70/0.9350**	<u>31.85/0.9306</u>
w/ event	17.32/0.5682	23.82/0.7690	28.76/0.8923	29.37/0.8795	<u>32.98/0.9374</u>	**33.94/0.9416**

Table 2: Analysis of the efficacy of event streams based on PSNR and SSIM. Best and second best indexs are marked in bold and underline.

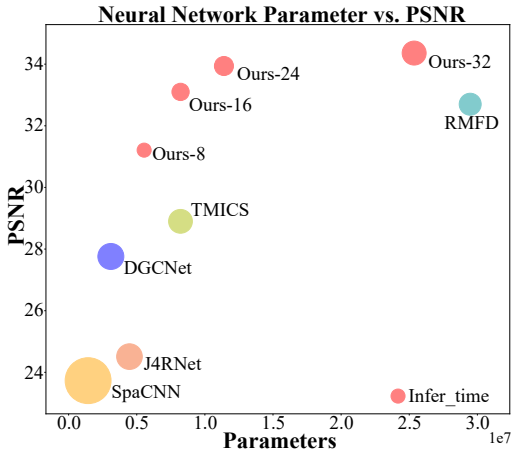


Figure 5: Network Parameter and Performance Comparisons: The vertical axis represents the PSNR achieved by each method on the *SynHeavy25* dataset, while the horizontal axis represents the parameter quantity of each method.

Ablation Studies

Effects of the Event Streams: We conduct experiments on six different de-raining methods to explore the effectiveness of event streams. To ensure a fair comparison, for the methods without the event processing module in their backbone networks, we concatenate the processed event streams onto intensity frames as input features for the whole network. For our method, we replace event streams with the intensity frames. As shown in Table 2, even without the event processing module, the introduction of event streams still results in an increase in the validation metric of de-raining methods by approximately 0.3dB/0.04. Especially for our method, the SSIM value decreases 0.110 but the PSNR value decreases 2.09dB. The reason behind this phenomenon is the lack of abundance high-frequency information provided by event streams. Our observations are further corroborated by Figure 1 in the **appendix**, which shows that without event streams, the moving masks generated by our network exhibit less accuracy, thereby leading to a large decrease in the PSNR value. This result indicates the efficacy of event streams in improving the de-raining performance.

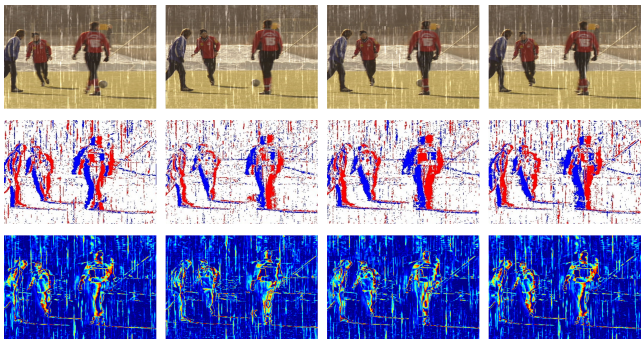


Figure 6: Features captured by the SME Module. From top to bottom are input frames, event representations and corresponding features.

Ablation Studies of Network Structures: To verify the effectiveness of our network modules, we conduct several ablation studies, and the results are presented in the **appendix**. The experiments demonstrate that incorporating the adaptive dictionaries (AD), Spatial Interaction (SI) and Temporal Interaction (TI) can enhance the de-raining performance of the network. Noticed that when we remove the AD Module, the whole process of **Updating** $\mathcal{R}/\mathcal{B}$ is replaced by a single U-Net module with the same parameters level.

Conclusion

In this paper, we explore the feasibility to introduce event streams into the VFD field. By modeling the dynamic information in event streams as a new prior constraint, we modify the existing optimization function. We also propose a deep unfolding algorithm and use a modular strategy to forward unfold the algorithm to construct the video de-raining network. What's more, we embed a well-designed spiking attention module within the framework to further excavate the spatial-temporal correlation. Extensive experiments show that our method achieves the SOTA on several synthetic and real-world benchmarks.

Acknowledgements

This work was supported by National Key R&D Program of China under Grant 2020AAA0105702, National Natural Science Foundation of China (NSFC) under Grants 62225207 and 62276243.

References

Aharon, M.; Elad, M.; and Bruckstein, A. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11): 4311–4322.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.; et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1): 1–122.

Cao, C.; Fu, X.; Zhu, Y.; Shi, G.; and Zha, Z.-J. 2022. Event-driven Video Deblurring via Spatio-Temporal Relation-Aware Network. In *IJCAI*.

Chen, J.; Tan, C.-H.; Hou, J.; Chau, L.-P.; and Li, H. 2018. Robust video content alignment and compensation for rain removal in a cnn framework. In *CVPR*, 6286–6295.

Duan, P.; Wang, Z. W.; Zhou, X.; Ma, Y.; and Shi, B. 2021. EventZoom: Learning to denoise and super resolve neuromorphic events. In *CVPR*, 12824–12833.

Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J. 2017. Removing rain from single images via a deep detail network. In *CVPR*, 3855–3863.

Fu, X.; Qi, Q.; Zha, Z.-J.; Zhu, Y.; and Ding, X. 2021. Rain streak removal via dual graph convolutional network. In *AAAI*, volume 35, 1352–1360.

Gallego, G.; Delbrück, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A. J.; Conradt, J.; Daniilidis, K.; et al. 2020. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1): 154–180.

Ge, C.; Fu, X.; and Zha, Z.-J. 2022. Learning Dual Convolutional Dictionaries for Image De-raining. In *ACM MM*, 6636–6644.

He, R.; Zheng, W.-S.; Tan, T.; and Sun, Z. 2013. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(2): 261–275.

Hu, Y.; Liu, S.-C.; and Delbruck, T. 2021. v2e: From video frames to realistic DVS events. In *CVPR*, 1312–1321.

Kim, S.; Park, S.; Na, B.; and Yoon, S. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *AAAI*, volume 34, 11270–11277.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, M.; Cao, X.; Zhao, Q.; Zhang, L.; and Meng, D. 2021. Online rain/snow removal from surveillance videos. *IEEE Trans. Image Processing*, 30: 2029–2044.

Li, M.; Xie, Q.; Zhao, Q.; Wei, W.; Gu, S.; Tao, J.; and Meng, D. 2018. Video rain streak removal by multiscale convolutional sparse coding. In *CVPR*, 6644–6653.

Li, W.; Chen, H.; Guo, J.; Zhang, Z.; and Wang, Y. 2022a. Brain-inspired multilayer perceptron with spiking neurons. In *CVPR*, 783–793.

Li, Y.; Chang, Y.; Yu, C.; and Yan, L. 2022b. Close the loop: a unified bottom-up and top-down paradigm for joint image deraining and segmentation. In *AAAI*, volume 36, 1438–1446.

Liu, J.; Yang, W.; Yang, S.; and Guo, Z. 2018. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *CVPR*, 3233–3242.

Mu, P.; Liu, Z.; Liu, Y.; Liu, R.; and Fan, X. 2021. Triple-Level Model Inferred Collaborative Network Architecture for Video Deraining. *IEEE Trans. Image Processing*, 31: 239–250.

Pan, L.; Scheerlinck, C.; Yu, X.; Hartley, R.; Liu, M.; and Dai, Y. 2019. Bringing a blurry frame alive at high frame-rate with an event camera. In *CVPR*, 6820–6829.

Rebecq, H.; Gehrig, D.; and Scaramuzza, D. 2018. ESIM: an open event camera simulator. In *Conference on robot learning*, 969–982.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019a. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, 3857–3866.

Rebecq, H.; Ranftl, R.; Koltun, V.; and Scaramuzza, D. 2019b. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6): 1964–1980.

Roy, K.; Jaiswal, A.; and Panda, P. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784): 607–617.

Sun, L.; Sakaridis, C.; Liang, J.; Jiang, Q.; Yang, K.; Sun, P.; Ye, Y.; Wang, K.; and Gool, L. V. 2022. Event-based fusion for motion deblurring with cross-modal attention. In *ECCV*, 412–428. Springer.

Sun, S.; Ren, W.; Li, J.; Zhang, K.; Liang, M.; and Cao, X. 2023. Event-aware Video Deraining via Multi-Patch Progressive Learning. *IEEE Transactions on Image Processing*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPs*, 30.

Wang, B.; He, J.; Yu, L.; Xia, G.-S.; and Yang, W. 2020a. Event enhanced high-quality image recovery. In *ECCV*, 155–171.

Wang, C.; Xing, X.; Wu, Y.; Su, Z.; and Chen, J. 2020b. Dcsfn: Deep cross-scale fusion network for single image rain removal. In *ACM MM*, 1643–1651.

Wang, H.; Xie, Q.; Zhao, Q.; Li, Y.; Liang, Y.; Zheng, Y.; and Meng, D. 2023. RCDNet: An interpretable rain convolutional dictionary network for single image deraining. *IEEE Trans. Neural Netw. Learn. Syst.*

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4): 600–612.

Weng, W.; Zhang, Y.; and Xiong, Z. 2021. Event-based video reconstruction using transformer. In *ICCV*, 2563–2572.

Yan, W.; Tan, R. T.; Yang, W.; and Dai, D. 2021. Self-aligned video deraining with transmission-depth consistency. In *CVPR*, 11966–11976.

Yan, W.; Xu, L.; Yang, W.; and Tan, R. T. 2022. Feature-Aligned Video Raindrop Removal With Temporal Constraints. *IEEE Trans. Image Processing*, 31: 3440–3448.

Yang, W.; Liu, J.; and Feng, J. 2019. Frame-consistent recurrent video deraining with dual-level flow. In *CVPR*, 1661–1670.

Yang, W.; Tan, R. T.; Feng, J.; Guo, Z.; Yan, S.; and Liu, J. 2019. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6): 1377–1393.

Yang, W.; Tan, R. T.; Feng, J.; Wang, S.; Cheng, B.; and Liu, J. 2021. Recurrent multi-frame deraining: Combining physics guidance and adversarial learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11): 8569–8586.

Yang, W.; Tan, R. T.; Wang, S.; Kot, A. C.; and Liu, J. 2022. Learning to Remove Rain in Video With Self-Supervision. *IEEE Trans. Pattern Anal. Mach. Intell.*

Yue, Z.; Xie, J.; Zhao, Q.; and Meng, D. 2021. Semi-supervised video deraining with dynamical rain generator. In *CVPR*, 642–652.

Zhang, K.; Gool, L. V.; and Timofte, R. 2020. Deep unfolding network for image super-resolution. In *CVPR*, 3217–3226.

Zhang, K.; Li, D.; Luo, W.; Ren, W.; and Liu, W. 2022. Enhanced Spatio-Temporal Interaction Learning for Video Deraining: Faster and Better. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1): 1287–1293.

Zhang, L.; Zhang, H.; Zhu, C.; Guo, S.; Chen, J.; and Wang, L. 2021a. Fine-grained video deblurring with event camera. In *MultiMedia Modeling*, 352–364.

Zhang, S.; Zhang, Y.; Jiang, Z.; Zou, D.; Ren, J.; and Zhou, B. 2020. Learning to see in the dark with events. In *ECCV*, 666–682.

Zhang, T.; Zeng, Y.; Zhao, D.; and Shi, M. 2018. A plasticity-centric approach to train the non-differential spiking neural networks. In *AAAI*, volume 32.

Zhang, X.; Liao, W.; Yu, L.; Yang, W.; and Xia, G.-S. 2021b. Event-based synthetic aperture imaging with a hybrid network. In *CVPR*, 14235–14244.

Zhu, A. Z.; Yuan, L.; Chaney, K.; and Daniilidis, K. 2019. Unsupervised event-based learning of optical flow, depth, and egomotion. In *CVPR*, 989–997.

Zhu, Y.; Xiao, Z.; Fang, Y.; Fu, X.; Xiong, Z.; and Zha, Z.-J. 2022. Efficient model-driven network for shadow removal. In *AAAI*, volume 36, 3635–3643.