Event-Based Motion Deblurring Using Task-Oriented 3D Gaussian Event Representations

Supplementary Material

Due to the limited space in the main paper, we provide additional details in this supplementary material. This supplementary material consists of the following sections.

Sec. 1 presents the detailed network architecture of our proposed approach, as well as implementation details of the event representation pipeline.

Sec. 2 provides supplementary experiments on the event representation.

Sec. 3 presents additional visual results on the GoPro dataset.

1. Method Details

1.1. Network Detailed Structure

We provide a more detailed network architecture in the method section of the main paper, as shown in Fig. 1. The original event stream is represented as $(B, 4, N)$, where 4 corresponds to (x, y, t, p) and N denotes the total number of events. We use several depthwise-separable 3D convolution blocks to extract high-level features, and employ a multiple MLP sampler to generate the parameters of the 3D Gaussian kernels in the format $(B, K, 10)$, where K is the number of kernels and 10 corresponds to $\mu_t, \mu_x, \mu_y, \sigma_t, \sigma_x, \sigma_y, \rho_{xt}, \rho_{yt}, \rho_{xy}$, and a bias term.

1.2. Implementation Details

We directly feed the raw event streams and their corresponding image samples into the network. A parameter-free neural module is used to dynamically convert event data into voxel-grid representations, fully leveraging the parallel computing capability of the GPU. This design allows us to embed the event preprocessing pipeline within the deblurring network itself, eliminating the need for CPU-based preprocessing into event frames, which is commonly adopted in existing methods and often incurs significant storage overhead. In addition, we replace full-image inference with patch-based inference to accommodate the end-to-end event representation pipeline.

1.2.1. Custom Collate Function

To embed the event preprocessing pipeline within the deblurring network, we implement a custom collate function to pad the number of events in each batch to a uniform length, thereby enabling efficient batch processing.

Given a batch $\mathcal{B} = \{(\mathbf{LQ}_b, \mathbf{GT}_b, \mathbf{EV}_b)\}_{b=1}^{B}$, where $\mathbf{LQ}_b$ and $\mathbf{GT}_b$ denote the blurry and sharp images, respectively, and $\mathbf{EV}_b \in \mathbb{R}^{4\times N_b}$ is the corresponding event sequence with variable length N_b, where the four dimensions correspond to x, y, t, p, the collation procedure standardizes all inputs before feeding them into the network.

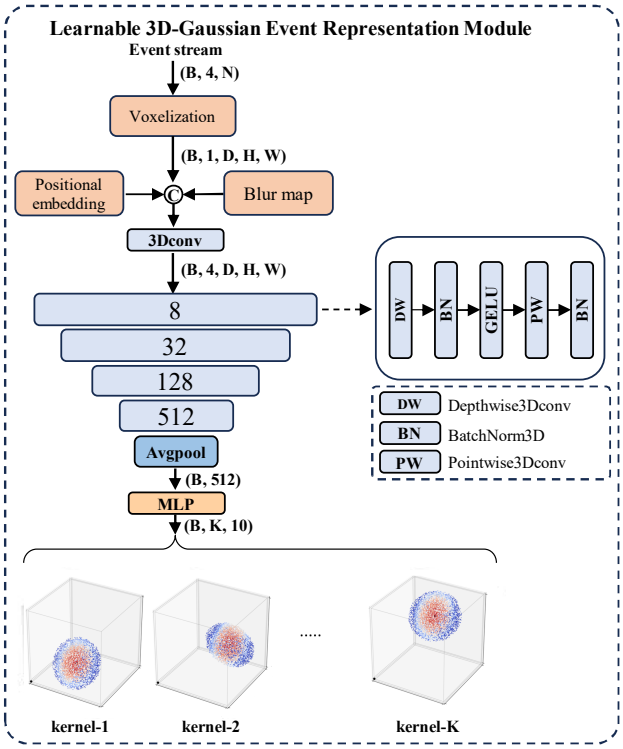

Figure 1. Details of the 3D Gaussian event representation module.

Image Stacking. Blurry and sharp images are stacked along the batch dimension:

$$\begin{aligned} \mathbf{LQ} &= \mathrm{Stack}(\mathbf{LQ}_1, \ldots, \mathbf{LQ}_B), \\ \mathbf{GT} &= \mathrm{Stack}(\mathbf{GT}_1, \ldots, \mathbf{GT}_B). \end{aligned} \tag{1}$$

Event Padding. To handle variable-length event streams, we pad each sequence to length $N_{\max} = \max_b N_b$ using a constant value:

$$\tilde{\mathbf{EV}}_b(i) = \begin{cases} \mathbf{EV}_b(i), & 1 \le i \le N_b, \\ 2, & N_b < i \le N_{\max}. \end{cases} \tag{2}$$

Output. The function returns $\mathbf{LQ}$ and $\mathbf{GT}$ in the format (B, C, H, W). The padded event sequences are stacked and permuted to match the network input format $(B, 4, N_{\max})$.

This ensures uniform tensor dimensions and enables efficient mini-batch processing. We use the padding value 2 because it is convenient to construct tensor-indexing masks such as $pos = [p == 1]$ and $neg = [p == 0]$, which can be used to quickly exclude invalid padded events.

1.2.2. Data Augmentation Function

To enhance the robustness and generalization capability of our model, we implement a comprehensive data augmentation pipeline that operates synchronously on both images and event streams. This ensures spatial and temporal consistency across all modalities during training.

Event Time Normalization. We first normalize the event timestamps to a standardized range:

$$t' = \frac{t - t_{\min}}{t_{\max} - t_{\min}} \times t_{\text{scale}}, \tag{3}$$

where $t_{\min}$ and $t_{\max}$ are the minimum and maximum timestamps in the event stream, respectively, and t_{scale} controls the temporal resolution scale.

Random Synchronized Cropping. Given the crop size P, image height H, and width W, we first randomly sample the coordinates of the top-left corner: top $\sim \mathcal{U}(0, H - P)$ and left $\sim \mathcal{U}(0, W - P)$, where bottom $=$ top $+ P$ and right $=$ left $+ P$. We then define the crop region and extract the corresponding patches:

$$\begin{aligned} \mathbf{LQ}' &= \mathbf{LQ}[\text{top} : \text{bottom}, \text{left} : \text{right}, :], \\ \mathbf{GT}' &= \mathbf{GT}[\text{top} : \text{bottom}, \text{left} : \text{right}, :], \\ \mathbf{EV}' &= \mathbf{EV}\left\{(x', y', t', p')\right\}, \end{aligned} \tag{4}$$

where $x' \in [\text{left}, \text{right})$ and $y' \in [\text{top}, \text{bottom})$.

Synchronized Flip and Rotation. We apply identical spatial transformations to maintain cross-modal alignment. The transformation type $k \sim \mathcal{U}(0, 7)$ determines the flipping and rotation operations:

$$(\mathbf{LQ}'', \mathbf{GT}'', \mathbf{EV}'') = \begin{cases} \text{Flip}(\mathbf{LQ}', \mathbf{GT}', \mathbf{EV}'), & k \geq 4, \\ \text{Rot}(\mathbf{LQ}', \mathbf{GT}', \mathbf{EV}'), & k < 4. \end{cases} \tag{5}$$

For horizontal flipping, the event coordinates are updated as:

$$x'' = (P - 1) - x'. \tag{6}$$

For rotation, applying the transformation $r = k \bmod 4$ times yields the iterative coordinate update:

$$(x'', y'') = (y', (P - 1) - x'). \tag{7}$$

This synchronized augmentation strategy preserves the spatial relationships between images and events while significantly increasing the diversity of training samples, thereby improving the generalization ability and robustness of the model to various spatial transformations.

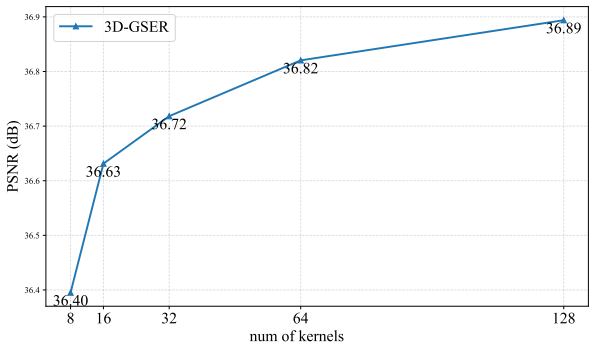

Figure 2. Ablation of the number of kernels on the GoPro dataset.

2. Additional Experiments

2.1. Ablation Study of the Number of Kernels

To fairly compare different numbers of kernels, we conduct an ablation study using our deblurring network (without the blur map and the second-stage refinement) as the baseline on the GoPro [1] dataset. As shown in Fig. 2, the PSNR increases monotonically with the number of kernels, indicating that more event frames generated by more kernels can preserve richer motion information from the raw events. However, as the number of kernels continues to increase, the performance gain gradually saturates. Therefore, it is important to strike a balance between computational cost and performance improvement.

Table 1. Ablation study on different event-based deblurring networks.

Method	Publication	Event Representation		
		SCER	DA	3D-GSER
EFNet[1]	ECCV'22	35.46	-	**36.48**
MAENet[2]	ECCV'24	-	36.07	**36.64**
MATNet[3]	AAAI'25	36.67	36.46	**36.93**

2.2. Ablation Study on Other Baselines

To verify the robustness of our method across different deblurring networks, we conduct an ablation study using different networks as baselines. Specifically, we use EFNet [1], MAENet [2], and MATNet [3], together with their corresponding event representation methods. The PSNR results on the GoPro [1] dataset show that our event representation generalizes well across different baselines and consistently outperforms the alternative methods.

3. Additional Visual Results

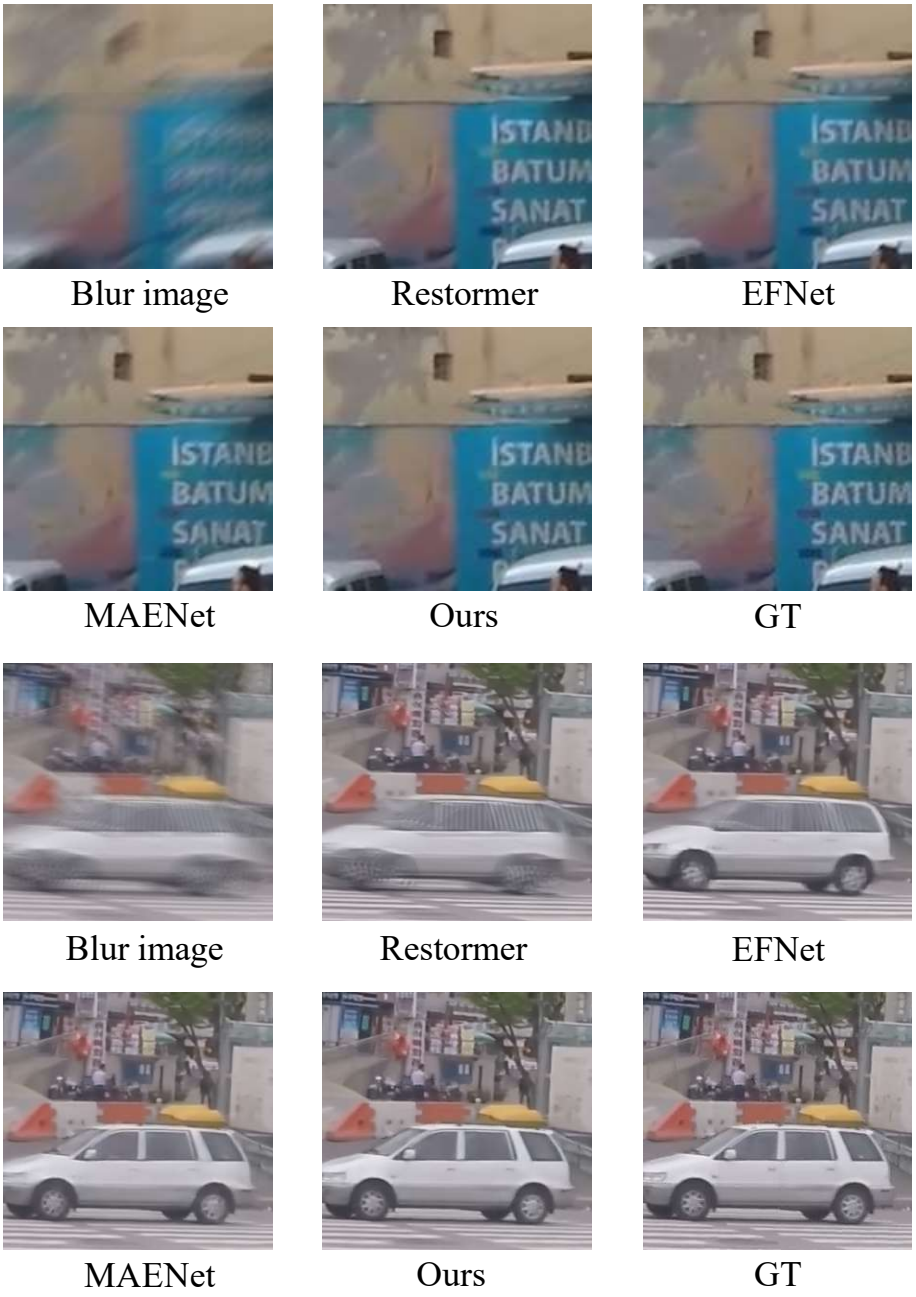

Figure 3. Visual results on the GoPro dataset.

References

[1] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *European conference on computer vision*, pages 412–428. Springer, 2022. 2

[2] Zhijing Sun, Xueyang Fu, Longzhuo Huang, Aiping Liu, and Zheng-Jun Zha. Motion aware event representation-driven image deblurring. In *European Conference on Computer Vision*, pages 418–435. Springer, 2024. 2

[3] Senyan Xu, Zhijing Sun, Mingchen Zhong, Chengzhi Cao, Yidi Liu, Xueyang Fu, and Yan Chen. Motion-adaptive transformer for event-based image deblurring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8942–8950, 2025. 2

Figure 4. Visual results on the GoPro dataset.