Latency Correction for Event-guided Deblurring and Frame Interpolation

Supplementary Material

Yixin Yang[1,2,†] Jinxiu Liang[1,2] Bohan Yu[1,2] Yan Chen[3] Jimmy S. Ren[3] Boxin Shi[1,2,*]

[1] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[2] National Engineering Research Center of Visual Technology, School of Computer Science, Peking University [3] SenseTime Research

{yangyixin93,cssherryliang,ybh1998,shiboxin}@pku.edu.cn yanchenace@gmail.com rensijie@sensetime.com

6. More latency curve details

Choice of function $\rho(\lambda)$. In Section 3.2[1], we introduce an average edge thickness function $s_\lambda(I)$ in Eq. (10). To obtain an λ-irrelevant representation, we integrate the right-side of the equation over λ with a function $\rho(\lambda)$ in Eq. (11). $\rho(\lambda)$ is a decreasing weighting function, which makes $s(I)$ pay more attention to smaller λ (sensitive to blurry edges) and less attention to greater λ (sensitive to sharp edges). We adopt the function $\rho(\lambda) = e^{-\lambda}$ in our experiments.

Integration of function $s(I)$. The integral in Eq. (11) is performed as a closed-form expression by integrating $\rho(\lambda)$ over λ. Define $\Phi(x) = \int_0^x \rho(\lambda)\mathrm{d}\lambda$, we obtained:

$$s(I) = \sum_{\boldsymbol{p}} \int_0^{\|\nabla I_{\boldsymbol{p}}\|_2} \rho(\lambda)\mathrm{d}\lambda = \sum_{\boldsymbol{p}} \Phi(\|\nabla I_{\boldsymbol{p}}\|_2). \quad (22)$$

With $\rho(\lambda) = e^{-\lambda}$, we have $\Phi(x) = 1 - e^{-x}$.

Choice of function $f(\cdot)$. In Section 3.3, we introduce a parameterized latency model based on the correlation between latency and illuminance L, described in Eq. (15). The function $f(\cdot)$ monotonically decreases with increasing illuminance L and is designed to map illuminance into logarithmic space, reflecting the logarithmic nature of brightness changes captured by events. To ensure positive outputs for illuminance within the range $[0, L_{\text{max}}]$, f is defined as:

$$f(L) = \ln \frac{L_{\text{max}}}{L + e}, \quad (23)$$

with Euler's number e ensuring smooth gradient transitions in the feasible regions. Empirically, setting $L_{\text{max}} = e^{10}$ simplifies the function to:

$$f(L) = 10 - \ln(L + e). \quad (24)$$

Camera response function considerations. Since illuminance L is not directly measurable in arbitrary scenes, we use the intensity of the blurry image B as a proxy to estimate the spatially-varying latency, resulting in $l_{\boldsymbol{p}} = \psi(\frac{B_{\boldsymbol{p}}}{2\Delta\tau})$, where $\psi(\frac{B_{\boldsymbol{p}}}{2\Delta\tau})$ is defined as:

$$\psi(\frac{B_{\boldsymbol{p}}}{2\Delta\tau}) = \sum_{i=0}^{K} a_i \cdot f(\frac{B_{\boldsymbol{p}}}{2\Delta\tau})^i, \quad (25)$$

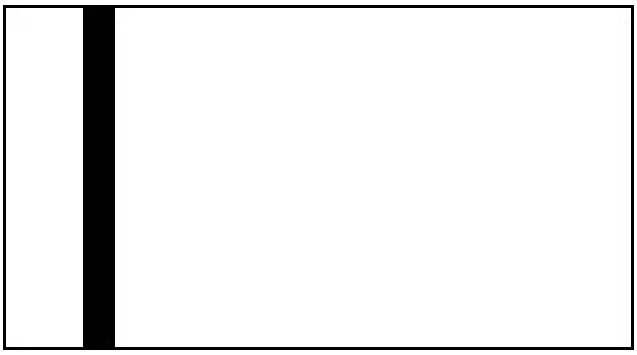

Figure 6. The strip video for capturing real events and generating ideal event. **GIF animation** could be displayed properly when viewed with Adobe Acrobat or KDE Okular.

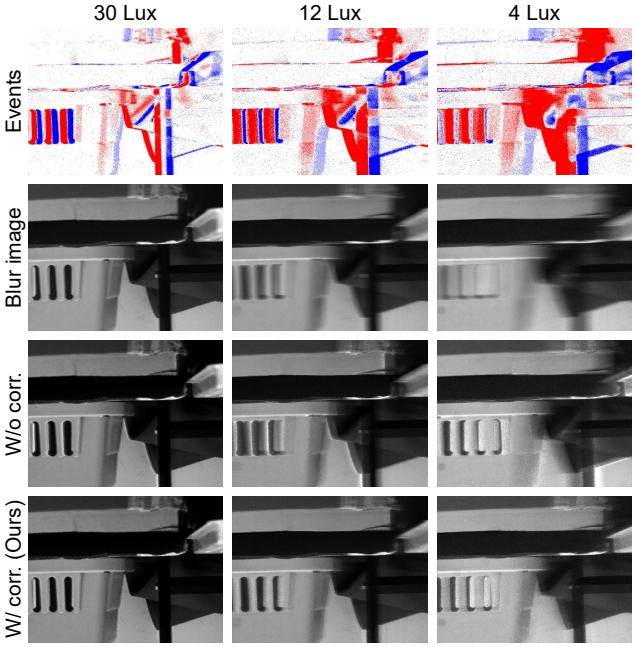

Figure 7. The complete results demonstrating the influence of latency on event-guided deblurring across different illumination conditions. From top to bottom: events; blur images; deblurring results guided by uncorrected events (Figure 1 (b)); and improved deblurring results employing events with latency corrected by our proposed method (Figure 1 (c)).

We do not explicitly incorporate the camera response function (CRF) here. Typically, CRF can be approximated as $\text{CRF}(\cdot) = \alpha(\cdot)^\gamma$ [4]. The intensity $B_{\boldsymbol{p}}$ at pixel $\boldsymbol{p}$ can

[†]This work is done during Yixin's internship at SenseTime.

[*]Corresponding author: shiboxin@pku.edu.cn

[1]All the cross-reference numbers (including figures, equations, and references) correspond to the main paper.

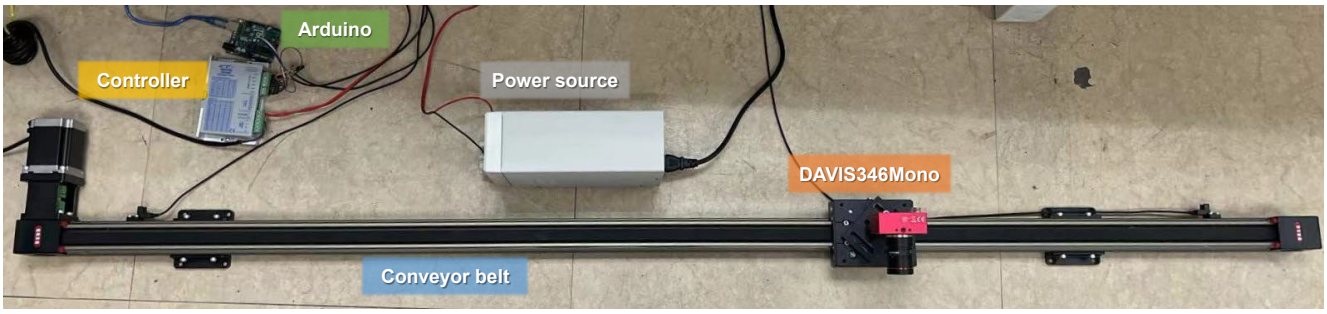

Figure 8. Illustration of the experimental setup for constant speed using a conveyor belt. The Arduino sends signals to the controller at a constant frequency, and the controller controls the speed and the direction of the belt at a constant speed.

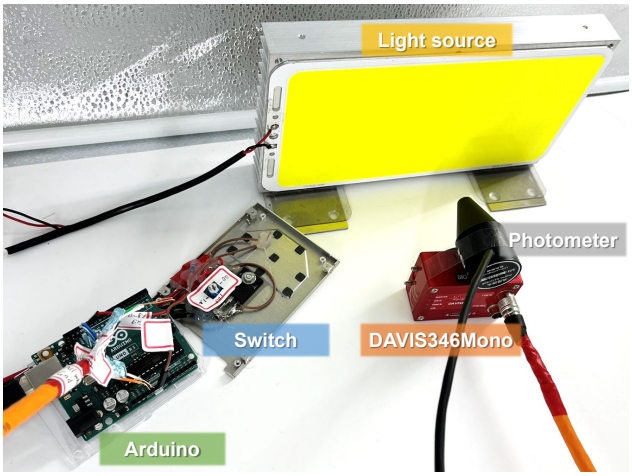

Figure 9. Illustration of the experimental setup for latency estimation in controlled environments, which includes an Arduino connected to a DAVIS346 via an orange cable, and a switch linked to an off-frame power source.

be approximated by the latent transient illuminance $L_{\boldsymbol{p}}$ at position $\boldsymbol{p}$, multipled by exposure time Δt, that is, $B_{\boldsymbol{p}} = \mathrm{CRF}(\Delta t L_{\boldsymbol{p}})$. Projecting B into the logarithmic space:

$$\ln B_{\boldsymbol{p}} \approx \ln \alpha + \gamma \ln \Delta t + \gamma \ln L_{\boldsymbol{p}}, \tag{26}$$

which means

$$\ln L_{\boldsymbol{p}} \approx \frac{1}{\gamma} \ln B_{\boldsymbol{p}} - \frac{1}{\gamma} \ln \alpha - \ln \Delta t. \tag{27}$$

In this context, the scaling factor $\frac{1}{\gamma}$ and the shift constant $-\frac{1}{\gamma} \ln \alpha - \ln \Delta t$ can be effectively approximated by the parameters of the proposed K-th order function when $K \geq 1$ in the curve Eq. (15). These terms, akin to the curve itself, are camera-specific parameters.

7. More experimental setup details

Figure 1. For Figure 1 (a), real events are captured from a monitor displaying a strip moving from left to right as illustrated in Figure 6. By disregarding the y-axis of the strip, it simulates a point moving along the x-axis from zero to infinity. Ideal events are generated from the strip video and the ideal event model (Eq. (1)). They are fewer then the real counterparts due to simultaneous triggering and event readout latency when capturing real events. Figure 1 (b) and (c) are captured indoors using a conveyor belt at constant speed as illustrated in Figure 8. To simulate lower illumination scenes, we increase the exposure time (keeping input illumination constant), ensuring a similar visual appearance. Complete results for Figure 1 (b) and (c), including input blurry images and events, are presented in Figure 7. The results highlight that poorer illumination conditions exacerbate latency's negative effect on deblurring performance, which benefits from the proposed method.

Figure 3. The experimental setups of the latency estimation results in Figure 3 (a) are depicted in Figure 9. All processes, including switching off the light and capturing events, were conducted in a darkroom.

Figure 5. Our real data were captured indoors using a DAVIS346Mono camera under various lighting conditions (full light, partial light, and no light) and aperture settings (ranging from 2.8 to 16). Frames and corresponding events were extracted from the recorded data, similar to the EDI model [19], and used to compile the results for the "W/o corr." condition.

8. Ablation of the loss functions

In order to validate the contribution of each component in Eq. (14) and Eq. (21), we disable Eq. (11), Eq. (12), Eq. (13), Eq. (19), and Eq. (20), respectively. Different components are compared on the EDI [19] method, which is the base module of the proposed latency correction method. As shown in Figure 10, the complete model provides the sharpest results among all the ablations with fewer artifacts and blurry remaining.

9. Computational efficiency

There are three parts of computation in the proposed method: constant latency, spatially-varying latency curve training, and spatially-varying latency testing. For constant latency, we search the constant latency l_{const} during the so-

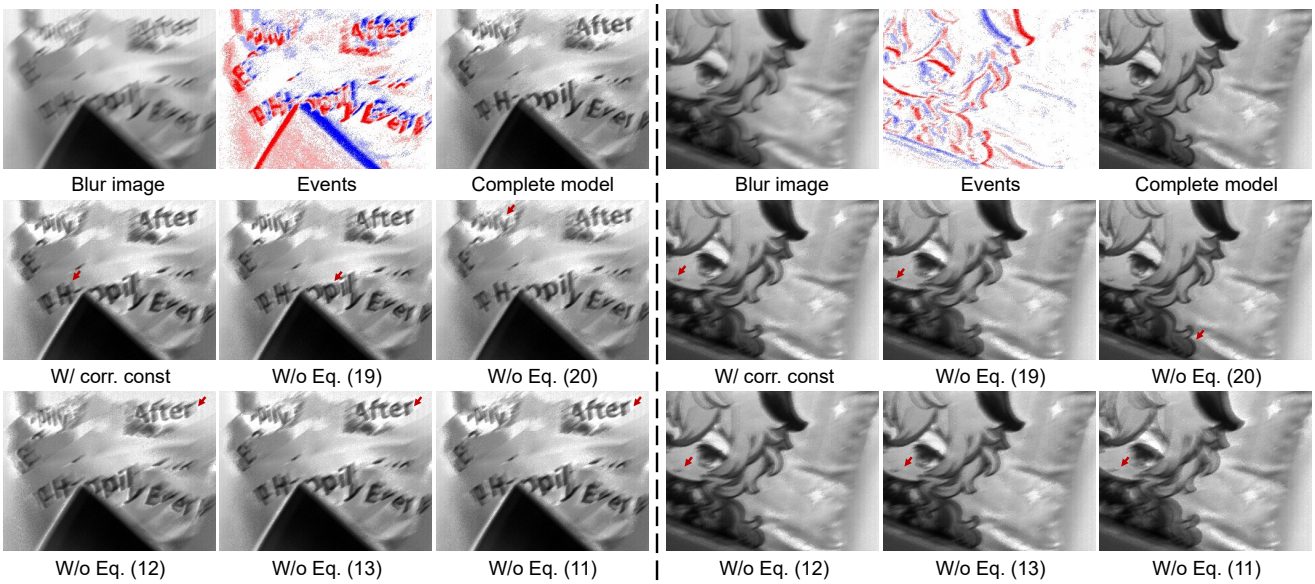

Figure 10. The qualitative comparison of ablation studies for different loss functions. As pointed out by the red arrows, all loss functions contribute to the deblurring quality in different ways.

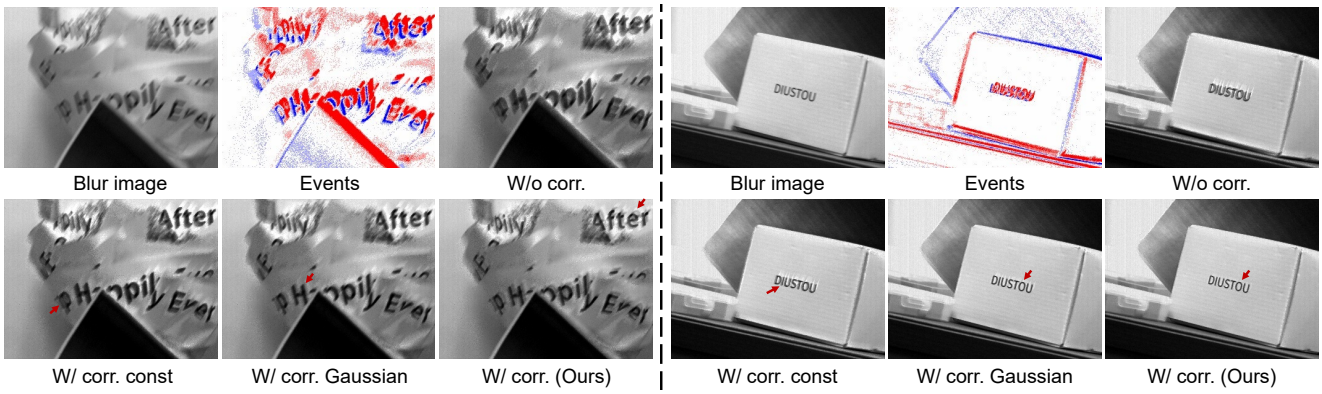

Figure 11. The qualitative comparison of the proposed method for two kinds of different differentiable event representation functions.

lution space (500 bins in our experiments), which takes 0.43s for each sample in average. For the training of the spatially-varying latency curve, we calculate l_{const} at first, then calculate loss functions and perform back-propagation. All those process takes 0.49s for one iteration, while it takes about 37min for $900 * 5 = 4500$ total iterations. For spatially-varying latency testing, we only calculate the spatially-varying latency for each pixel and obtain corresponding event streams, which takes 0.1s for each sample in average.

10. Alternative function to Eq. (16)

In Section 3.3, we propose reforming the event integral to be latency-differentiable by employing a piece-wise constant function in Eq. (16). However, the integration in Eq. (18) is not continuous at the time of $\{t_k\}_{k=1}^{N}$. An alternative to Eq. (16) is its softened version using Gaussian function, which gives us nonzero gradients everywhere. Therefore, we formulate and implement the Gaussian alternative function of Eq. (16). Denote an event $e_k = (\boldsymbol{p}_k, t_k, \sigma_k)$ as:

$$E_{\boldsymbol{p}}^{(k)}(s) = \frac{\sigma_k}{\sqrt{2\pi}\phi} \exp\{-\frac{(s-t_k)^2}{2\phi^2}\}. \tag{28}$$

By defining the integration of the Gaussian function as:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \text{d}t, \tag{29}$$

we can obtain the integration of $E_{\boldsymbol{p}}^{(k)}(s)$ for $-\infty < s < t$ is:

$$\int_{-\infty}^{t} E_{\boldsymbol{p}}^{(k)}(s) = \frac{\sigma_k}{2}(\text{erf}(\frac{t-t_k}{\sqrt{2}\phi}) + 1), \tag{30}$$

where ϕ is a hyperparameter determining the shape of the approximation. We set $\phi = 0.001$ in our experiments. Then the right side of Eq. (18) becomes:

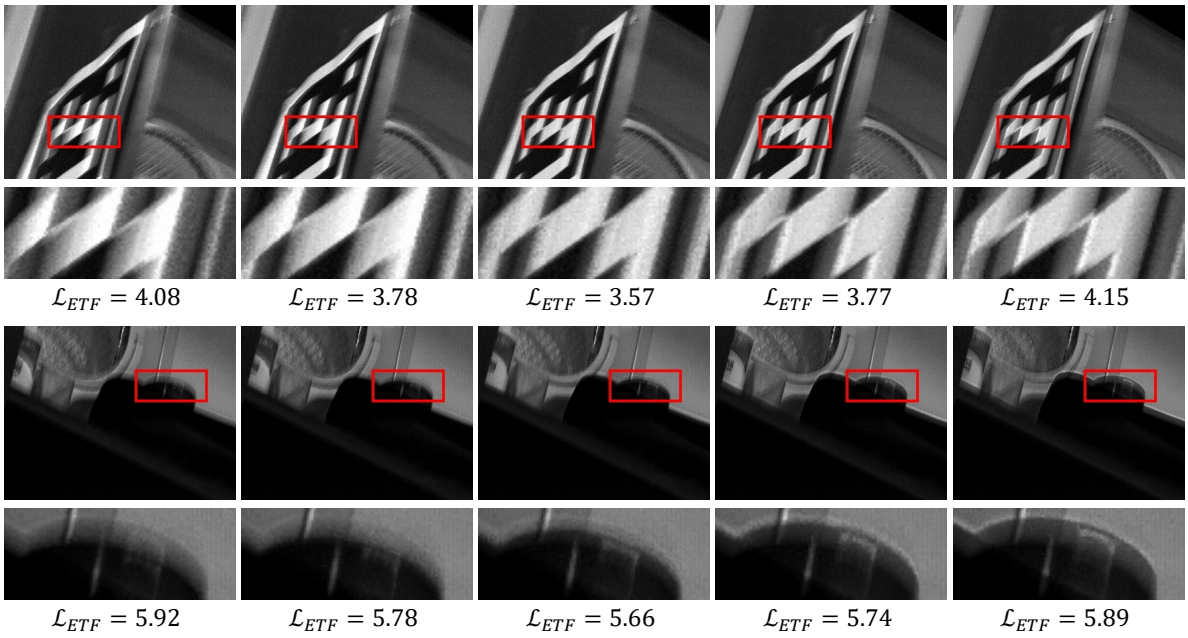

Figure 12. Illustration of the correlation between event-based temporal fidelity and the sharpness of latent images reconstructed from latency-corrected events and blurry images on real data. It shows that images with lower sharpness and thicker edges exhibit higher $\mathcal{L}_{\text{ETF}}$ values. These examples validate the efficacy of $\mathcal{L}_{\text{ETF}}$ as a metric for quantifying the sharpness profile of latent images I, reconstructed from latency-corrected events $l_{\boldsymbol{p}}$ and the corresponding blurry images B.

$$\sum_{k=1}^{N} \frac{\sigma_k}{2}(\text{erf}(\frac{t - l_{\boldsymbol{p}} - t_k}{\sqrt{2}\phi}) - \text{erf}(\frac{\tau - l_{\boldsymbol{p}} - t_k}{\sqrt{2}\phi})),$$

which varies continuously as the value of latency changes.

We train the piece-wise constant function (denoted as "W/ corr. (Ours)") and the Gaussian alternative function (denoted as "W/ corr. Gaussian") with the same setting as detailed in Section 3.4. The qualitative comparison is shown in Figure 11. The results show that the Gaussian alternative function shows comparable performance with the proposed piece-wise constant function in artifact suppression and image sharpness recovery.

11. Useful illuminance range for correction

In low-illumination levels, the event cameras also suffer from different kinds of heavy noise which is the same as the traditional image sensors. The proposed method aims at alleviating the latency effects under low-light conditions, and it may suffer from noises brought by low-light conditions. Therefore, evaluating the useful illuminance range for latency correction is important.

In addition to illuminance, the aperture is also an important factor that influences the luminous flux as well as latency. We evaluate the proposed method using the DAVIS346Mono, equipped with a fixed F8.0 aperture lens, under a range of illuminations from 0.01 Lux to 100 Lux (a typical normal indoor lighting[2]). A neutral density filter with 1% light transmittance is employed on the lens to mitigate the limitations in photometer accuracy below 1 Lux.

We adopt a similar system as Figure 9 to capture real data under different illumination level by adjusting the luminance of the light source. The results are shown in Figure 13, which indicate that latency becomes less significant above 50 Lux, while the quality of blurry image limits the ability of latency correction under 0.01 Lux. The results demonstrate its effectiveness across these varied lighting conditions (the ranges might vary in different setups).

12. More results on real data

Evaluating event-based temporal fidelity. We design $\mathcal{L}_{\text{ETF}}$ to measure the sharpness of latent images I reconstructed from events corrected for latency $l_{\boldsymbol{p}}$ and their corresponding blurry image B. To demonstrate that $\mathcal{L}_{\text{ETF}}$ accurately reflects image sharpness, Figure 12 presents examples with their respective $\mathcal{L}_{\text{ETF}}$ values labeled beneath each image. The amplifier ϵ in Equation (11) is set to 10.

Visual comparisons of latency correction results. We provide more visual comparisons of the proposed latency correction methods in Figure 14, Figure 15 , Figure 16, Figure 17, and Figure 18. The results demonstrate the improved performance of deblurring and interpolation at-

[2]Seeing in Extra Darkness Using a Deep-Red Flash, CVPR'21, Fig. 1

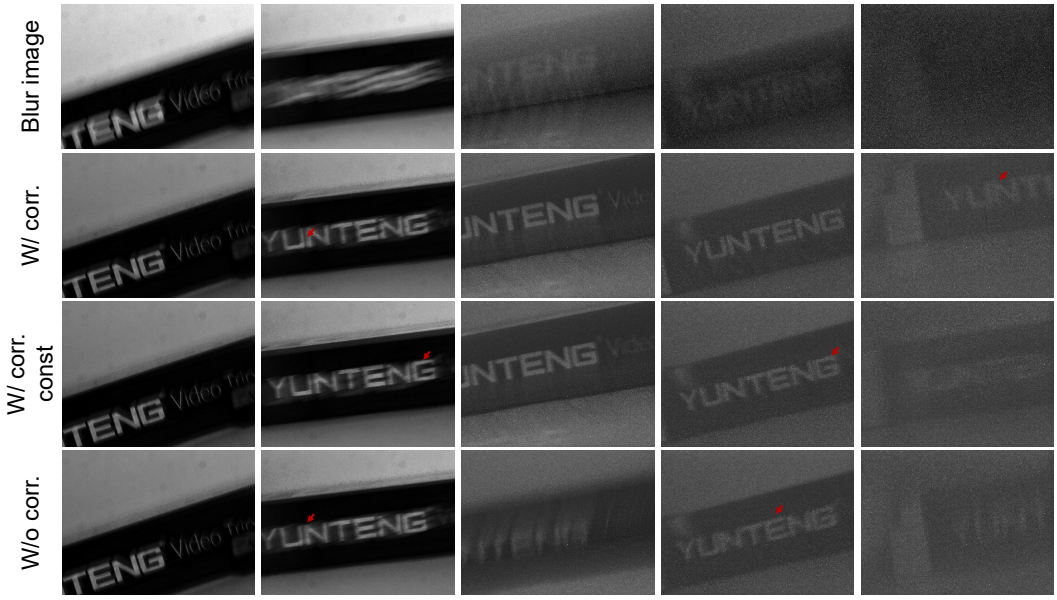

Figure 13. The results under different illuminance. "$\times 1\%$" means the camera equipped with a neutral density filter.

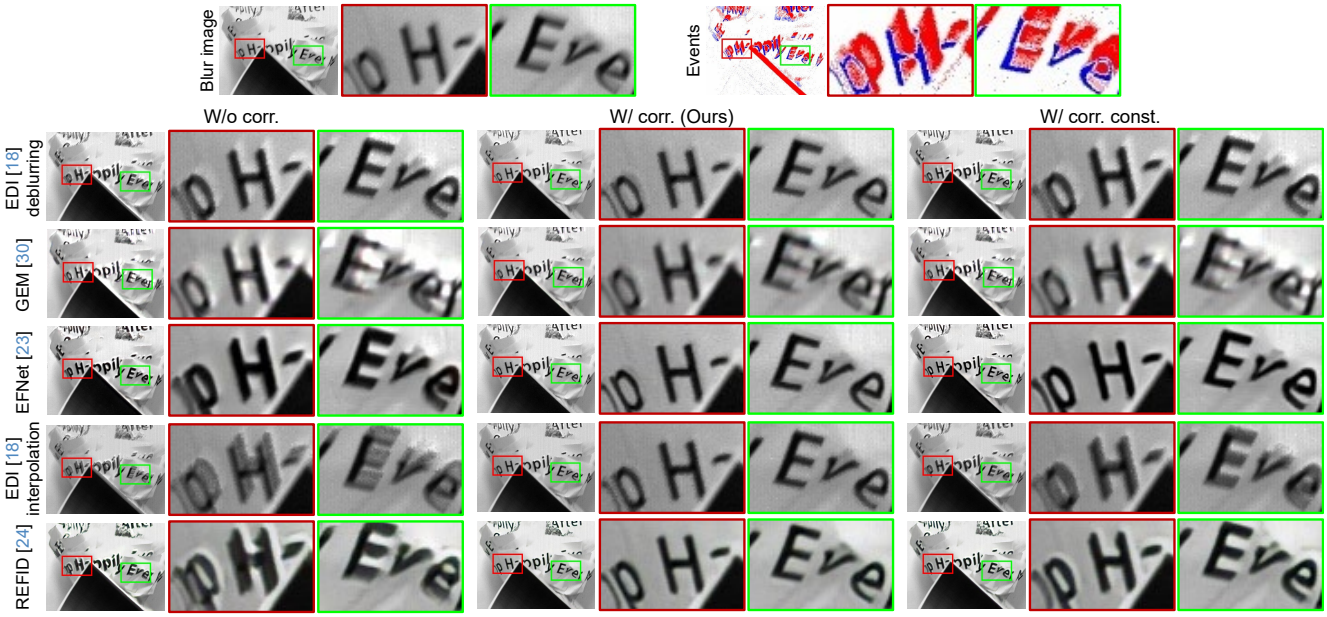

Figure 14. The qualitative comparison on real data. We correct the latency of events by constant latency correction ("W/ corr. Const.") and spatially-varying latency correction ("W/ corr."). The proposed method can effectively suppress the artificial thick edges generated by learning-based algorithms [24, 25], with the spatially-varying correction showing superior edge suppression compared to the constant correction. For other methods [19, 31], results come from latency correction present fewer artifacts and sharper appearance. Notably, the spatially-varying correction improves over constant latency correction, yielding more natural and refined results.

tributable to our proposed latency correction methodology. They shows that the proposed latency correction method can explicitly improve the performance of deblurring and interpolation methods, and the spatially varying latency correction provides more accurate latency and results in better appearance. Notably, the proposed spatially varying latency correction method yields more precise latency estimations and consequently, visibly improved image quality.

The non-learning EDI model [19], applied to both deblurring and interpolation, shows the most notable improvements, attributed to its sensitivity to the precision of event timestamps. Learning-based methods like EFNet [24] and

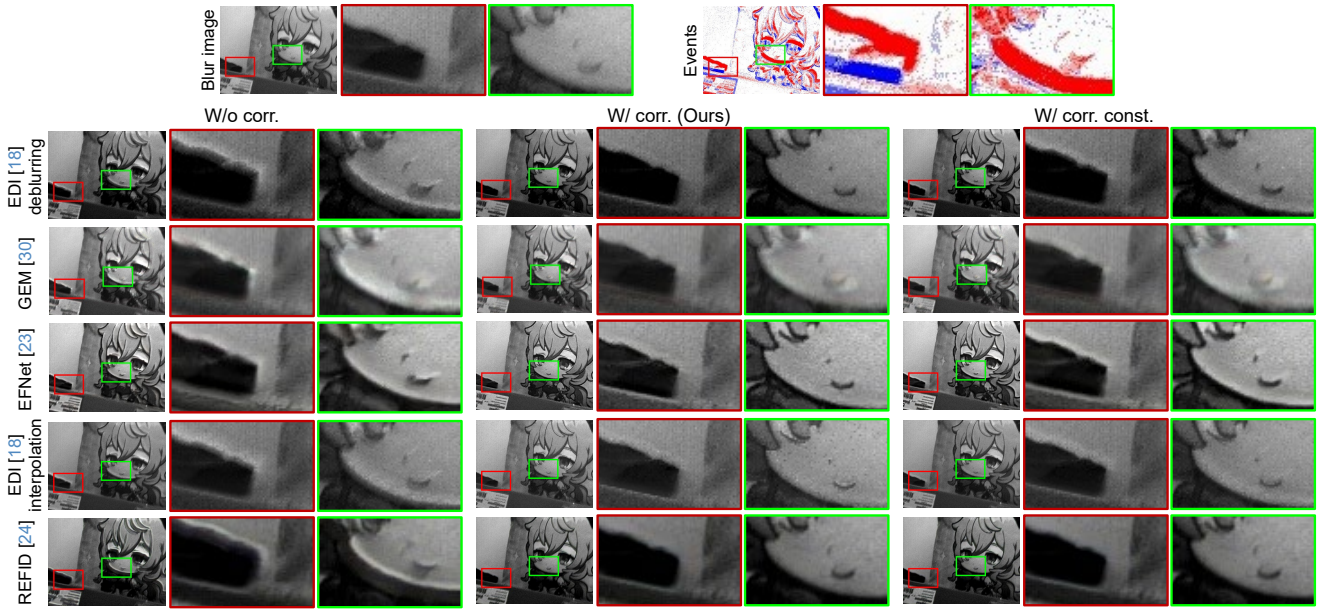

Figure 15. The qualitative comparison on real data. We correct the latency of events by constant latency correction ("W/ corr. Const.") and spatially-varying latency correction ("W/ corr."). The different width of the light strips in high contrast edges shows the different number of inaccuracy events.

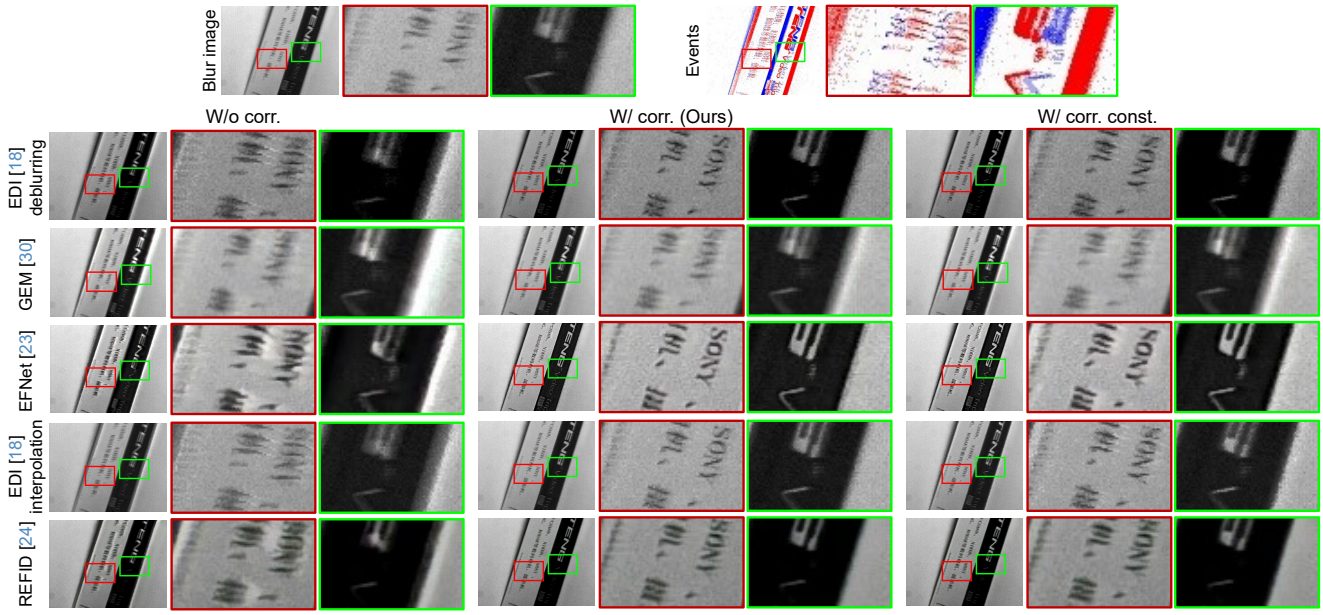

Figure 16. The qualitative comparison on real data. We correct the latency of events by constant latency correction ("W/ corr. Const.") and spatially-varying latency correction ("W/ corr."). As shown in the words in the green boxes and the contours in the red boxes, the proposed method notably improves deblurring and interpolation results, effectively restore finer details.

REFID [25], though inherently more resilient to timestamp inaccuracies, also benefit from the proposed latency correction method. Interestingly, GEM [31], trained on diverse real-world data under various lighting conditions, already incorporates latent latency effects in its training dataset.

The results shows that with the proposed method, classic event-guided deblurring [19, 24, 31] and interpolation methods [19, 25] can achieve improved robustness to latency in events. This mitigates the necessity for extensive redesign or retraining with large-scale datasets across various illumination conditions. It paves the way for the development of more refined deblurring and frame interpolation techniques, focusing on addressing fundamental challenges.

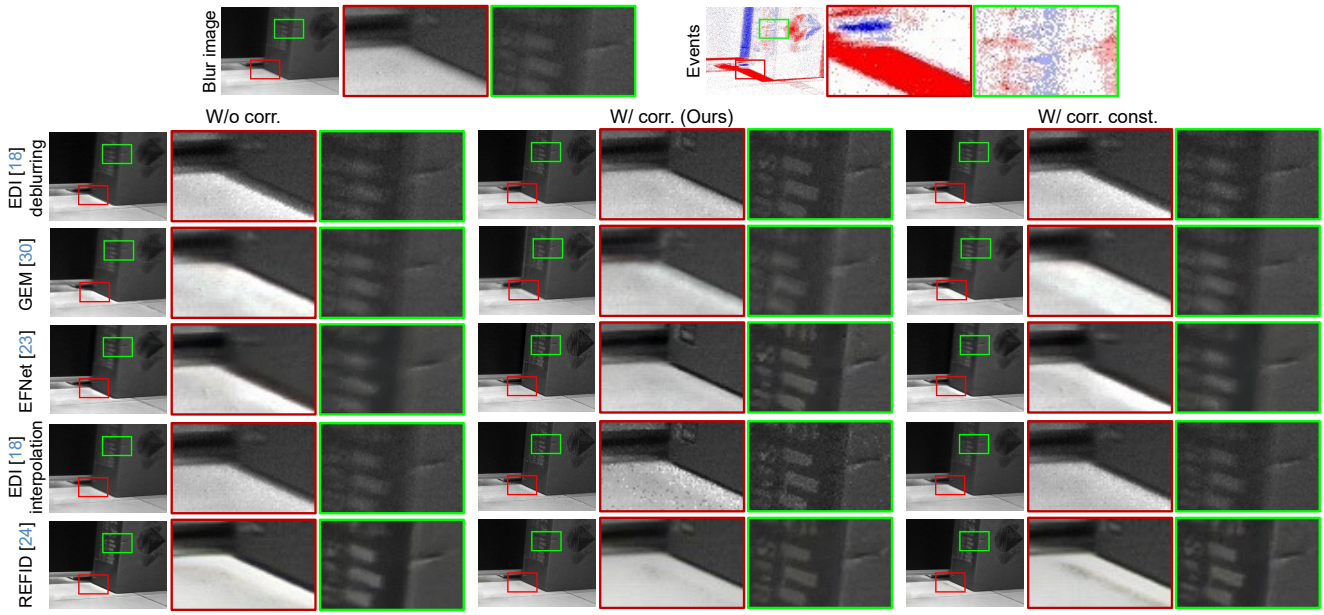

Figure 17. The qualitative comparison on real data. We correct the latency of events by constant latency correction ("W/ corr. Const.") and spatially-varying latency correction ("W/ corr."). All the methods are tested with the same input. The proposed event latency correction method effectively mitigates latency effects, improving the performance of existing deblurring and interpolation techniques in low-light conditions.

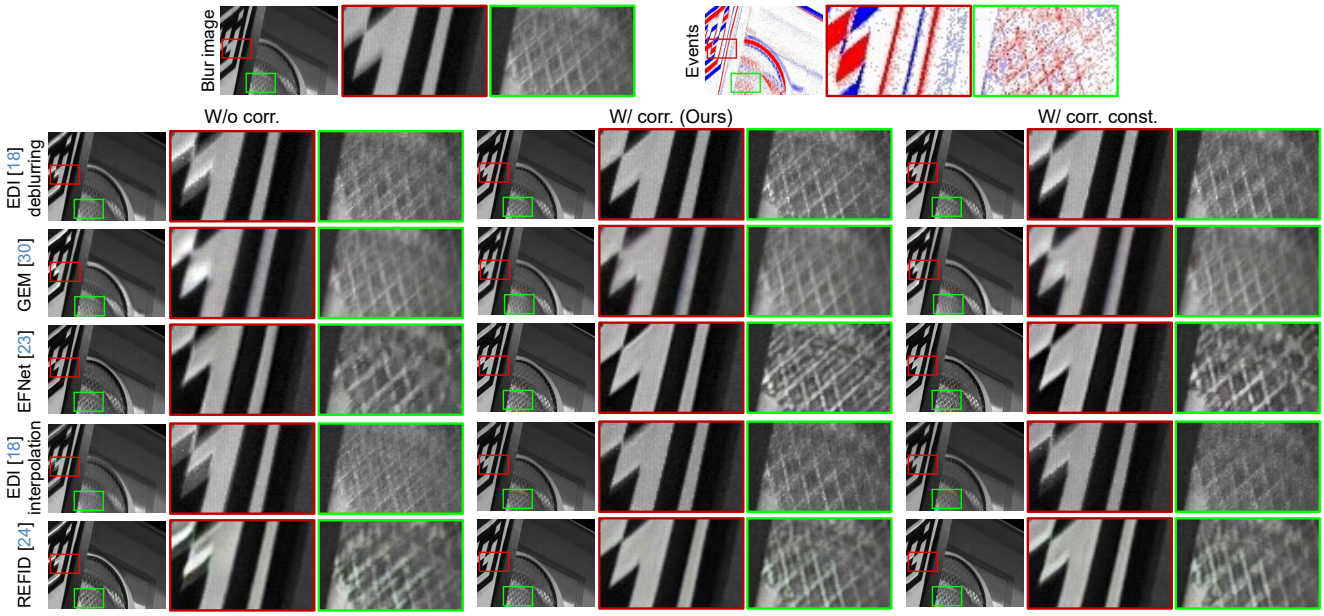

Figure 18. The qualitative comparison on real data. We correct the latency of events by constant latency correction ("W/ corr. Const.") and spatially-varying latency correction ("W/ corr."). All the methods are tested with the same input. The proposed event latency correction method effectively mitigates latency effects, improving the performance of existing deblurring and interpolation techniques in low-light conditions.