Supplementary Material: DVS-Voltmeter: Stochastic Process-based Event Simulator for Dynamic Vision Sensors

Songnan Lin[1⋆], Ye Ma[2⋆], Zhenhua Guo[3], and Bihan Wen[1⋆⋆]

[1] Nanyang Technological University, 50 Nanyang Ave, Singapore
[2] McGill University, Montreal, Canada
[3] Alibaba Group, Hangzhou, China

1 Overview

In this supplementary material, we first provide the implementation details of the proposed model calibration in Section 2. We present further quantitative analysis and discussion on the fidelity of event timestamps, the running time, and the temperature effect in Section 3. We show additional qualitative comparison among real events, our results and other simulated ones in Section 4. Finally, we provide more visual comparisons on two applications, *i.e.* semantic segmentation and intensity-image reconstruction, on real event data in Section 5 and Section 6, respectively.

2 Implementation Details of Model Calibration

To train deep neural networks generalizing to a specific DVS camera, it is necessary to accurately calibrate the parameters in the proposed event model and generate realistic events for this DVS camera. The parameters, including k_1, k_2,..., k_6, are related to sensors and environment (Please see Eq. (10) in the original paper for details). Ideally, we can look up the specification of the camera and conduct statistical experiments on noise effects to determine the parameters, similar to V2E [3]. However, the statistical experiments need complex equipment, and thus, are difficult to implement. In this paper, we provide a calibration method to get the parameters from a sequence of intensity frames and corresponding events.

Specifically, for every event recorded between two adjacent frames F_i and F_{i+1}, we can get the brightness conditions when the event occurs, including an approximated brightness $\bar{L} = (F_i + F_{i-1})/2$ and a brightness changes $\Delta L = F_i - F_{i-1}$. As intensity frames have limited dynamic range resulting in the loss of details, the events relevant to the black and white regions of the frames are removed from calibration. Furthermore, we collect the time interval τ between this event and the last event triggered at the same pixel. Then, given a specific

⋆ Equal contribution
⋆⋆ Corresponding author: bihan.wen@ntu.edu.sg

pair of $\bar{L}$ and ΔL, we can find the histogram of τ with a form of inverse Gaussian function as shown in Fig. 1 and fit it using maximum-likelihood estimation.

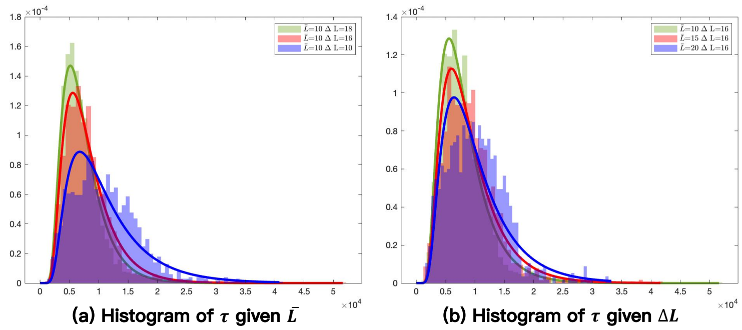


Fig. 1. Distribution of the time interval τ between two adjacent events at the same pixel. τ has an inverse Gaussian distribution / Lévy distribution. (a) The larger brightness change ΔL, the more compressed the distribution. (b) The larger average brightness $\bar{L}$, the more spread out the distribution. The statistical result is consistent with the proposed model.

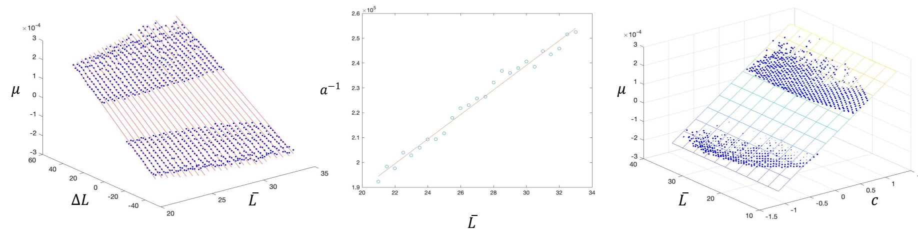
After obtaining the coefficients of the inverse Gaussian-distributed τ, we calculate the drift parameter μ and scale parameter σ of the Brownian motion-based event model for each pair of $\bar{L}$ and ΔL from Eq (13) in the original paper. According to Eq (10)(11) in the original paper,

$$\begin{aligned} \mu &= \frac{k_1}{\bar{L}+k_2} k_{dL} + k_4 + k_5 \bar{L} \\ \sigma &= \frac{k_3}{\bar{L}+k_2} \sqrt{\bar{L}} + k_6, \end{aligned} \tag{1}$$

in which, $k_{dL} = \frac{\Delta L}{\Delta t}$. Theoretically, given a set of $\{\mu_m, \sigma_m, \bar{L}_m, \Delta L_m\}, m \in N$, parameters k_1, k_2, ..., k_6 can be calculated by multivariable nonlinear regression. In practice, we find that it is hard to get an accurate estimation, so that we adopt linear regression several times and obtain the parameters sequentially.

As for the drift parameter μ in Eq. (1), given a specific $\bar{L}_n$, μ gets linear relation with the brightness change ΔL formulated as $\mu = a(\bar{L}_n)\Delta L + b(\bar{L}_n)$, which is consistent with the observation in Fig. 2 (a). We estimate robust regression coefficients for the linear model and obtain $\{a_n, b_n, \bar{L}_n\}$. Then, there is a linear relation between a^{-1} and $\bar{L}$ and thus, we obtain k_2 by linearly fitting $a^{-1} = k_1^{-1}\bar{L} + k_1^{-1}k_2$ as shown in Fig. 2 (b). After that, we set $c = (\bar{L}+k_2)^{-1}k_{dL}$, and estimate robust regression coefficients for the multivariable linear model $b = k_1 c + k_5 \bar{L} + k_4$ as shown in Fig. 2 (c).

As for the σ-related parameters in Eq. (1), we find that the auto-calibration method is limited by the quality of APS, image quantization, and the assumption of constant brightness changes in our model, which introduces large errors on



(a) $\mu = a(\bar{L})\Delta L + b(\bar{L})$ (b) $a^{-1} = k_1{}^{-1}\bar{L} + k_1{}^{-1}k_2$ (c) $\mu = k_1 c + k_5\bar{L} + k_4$

Fig. 2. μ-related parameter estimation. We conduct 3-time robust linear regression to estimate μ-related parameters sequentially. (a) Linear relation between μ and the brightness change ΔL given a specific $\bar{L}$. (b) k_2 is obtained by fitting the linear model between a^{-1} and $\bar{L}$. (c) k_1, k_5, k_4 are obtained by multivariable linear regression, where $c = (\bar{L} + k_2)^{-1} k_{dL}$.

σ-related parameters in challenging scenes, such as high dynamic range and fast motion. Therefore, we calibrate the σ-related parameters manually.

3 Quantitative Analysis and Discussion

To further validate the effectiveness of the proposed DVS-Voltmeter, we compare DVS-Voltmeter with existing methods [4,3] in terms of the fidelity of timestamps, the running time and the temperature effect.

Table 1. Quantitative comparison in terms of the Wasserstein distance of time intervals (ms) between real events [1] and simulated ones as well as the running time (ms per frame pair) with the size 346×260.

Methods	Vid2E [4]	V2E [3]	Ours
Wasserstein Distances (ms)	228.9	226.6	47.9
Running time (ms per pair)	0.4	14.7	15.5

3.1 Fidelity of Event Timestamps

To quantitatively analyze the fidelity of event timestamps, for each event, we calculate the time interval τ after the last event triggered at the same pixel, and then measure the Wasserstein distance of τ. The comparison between real event dataset [1] and the simulated ones is provided in Table 1. Our DVS-Voltmeter produces much more realistic event data compared with the existing simulators.

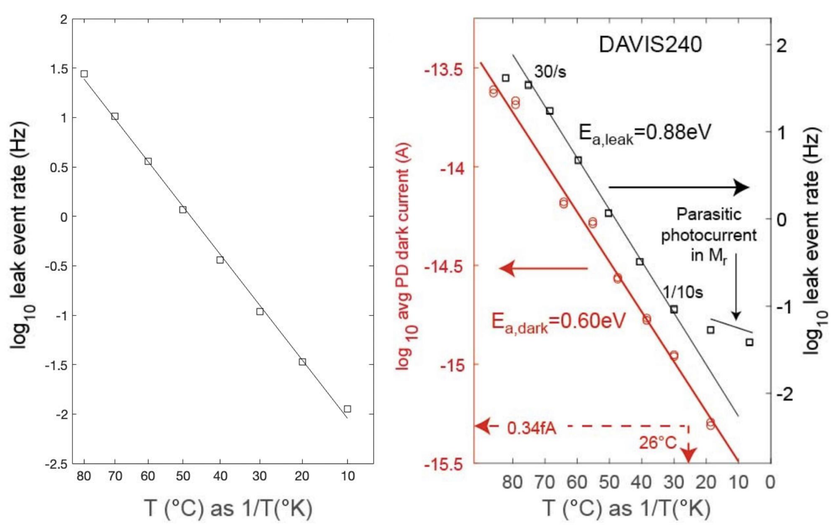
3.2 Running time

Table 1 shows the average time cost based on the video frames with a size of 346×260. Vid2E [4] runs on a RTX A5000 GPU, while V2E [3] and our DVS-Voltmeter run on a 3.7GHz AMD Ryzen Threadripper 3970X CPU. DVS-Voltmeter is comparable to V2E [3], while less efficient than Vid2E [4] due to its complex random number sampling and CPU implement. We leave GPU implement of DVS-Voltmeter in the future.

3.3 Temperature Effect

To quantify the effectiveness of the proposed model, we conduct an experiment to compare the temperature effect of our simulated events with the statistical results from real DVS. We adjust the temperature-related parameter k_4 in the proposed model according to Eq. (4) in the original paper, and continuously feed a single dark image into the simulator to measure the noise rate.

Fig. 3 shows leak event rate as a function of temperature plotted as the log of the quantity versus reciprocal of absolute temperature. The temperature axis is labeled with centigrade. As our simulator is designed based on the DVS pixel circuit and incorporates a noise term to model the noises from temperature, it naturally represents the distribution of real events. As shown in Fig. 3(a), the number of leak events increases with the temperature, similar to the real statistical analysis in Fig. 3(b).



(a) Ours **(b) Real**

Fig. 3. Comparisons on the effects of temporature. We continuously feed a single dark image into our simulator and measure the noise rate at different temperature. (a)(b) are the results of our simulator and real DAVIS240C data provided by [7]. Our noise rate is similar to the real statistical analysis.

4 More Qualitative Comparison

We exhibit additional side-by-side comparison of real public event datasets [6,2] and the simulated reproductions from Vid2E [4], V2E [3] and our DVS-Voltmeter. Fig. 4 to Fig. 7 show the events between four adjacent frames in forms of spatiotemporal event clouds, exponential time surface [5] with exponential decay 3.0ms and probability density function histograms of time intervals.

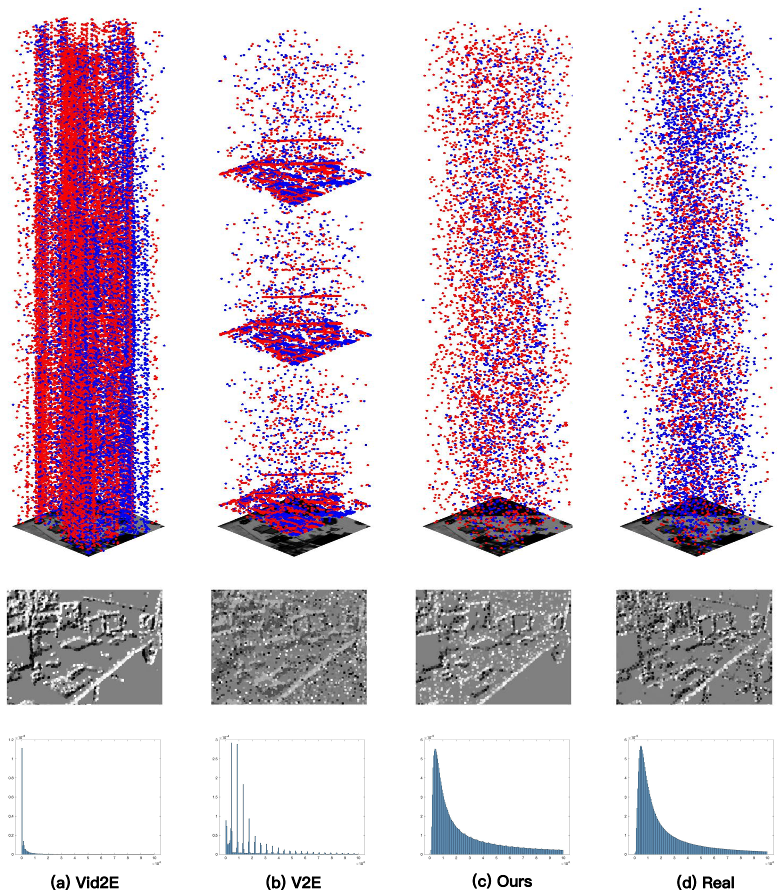


Fig. 4. Comparison on DAVIS240C 'office_zigzag' data [6]. (a)(b)(c) Simulated reproduction from Vid2E [4], V2E [3], our DVS-Voltmeter. (d) real DAVIS240C data. We illustrate 3D clouds, 2D time surfaces, and probability density function histograms of event data from top to bottom. The color pair (red, blue) represents the polarity (1,-1) of events. The proposed DVS-Voltmeter gains more randomness and the generated events resemble real data.

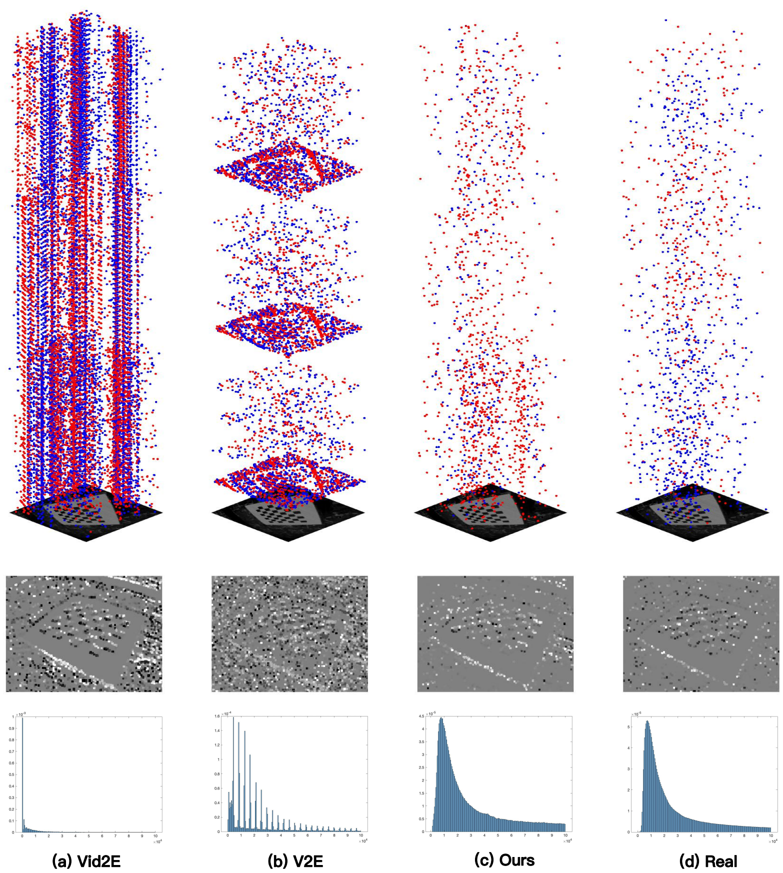


Fig. 5. Comparison on DAVIS240C 'calibration' data [6]. (a)(b)(c) Simulated reproduction from Vid2E [4], V2E [3], our DVS-Voltmeter. (d) real DAVIS240C data. We illustrate 3D clouds, 2D time surfaces, and probability density function histograms of event data from top to bottom. The color pair (red, blue) represents the polarity (1,-1) of events. The proposed DVS-Voltmeter gains more randomness and the generated events resemble real data.

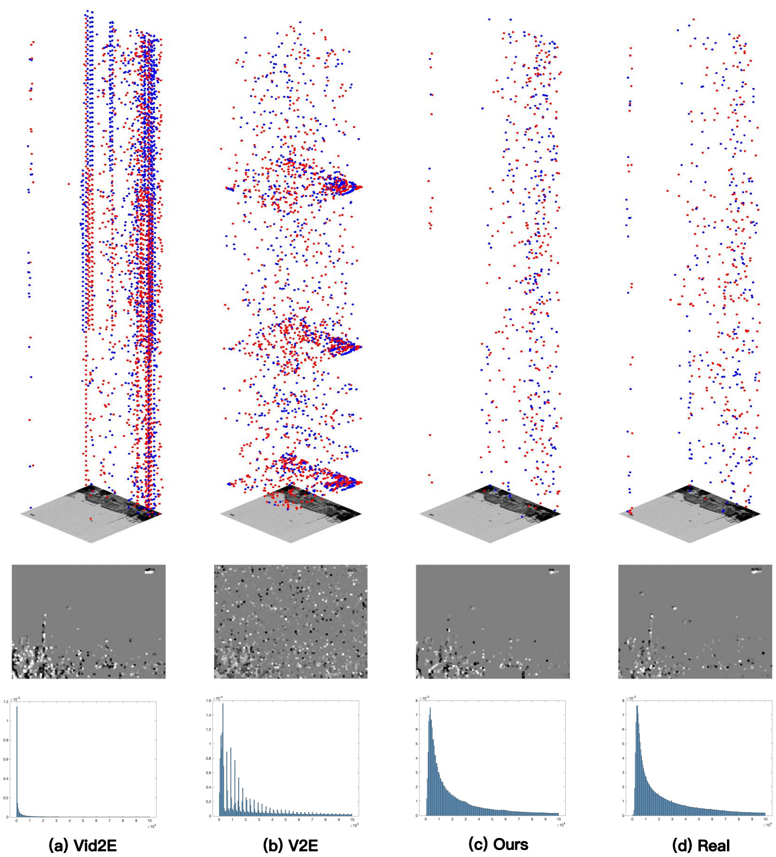


Fig. 6. Comparison on DAVIS346 'rec1487339175' data [2]. (a)(b)(c) Simulated reproduction from Vid2E [4], V2E [3], our DVS-Voltmeter. (d) real DAVIS346 data. We illustrate 3D clouds, 2D time surfaces, and probability density function histograms of event data from top to bottom. The color pair (red, blue) represents the polarity (1,-1) of events. The proposed DVS-Voltmeter gains more randomness and the generated events resemble real data.

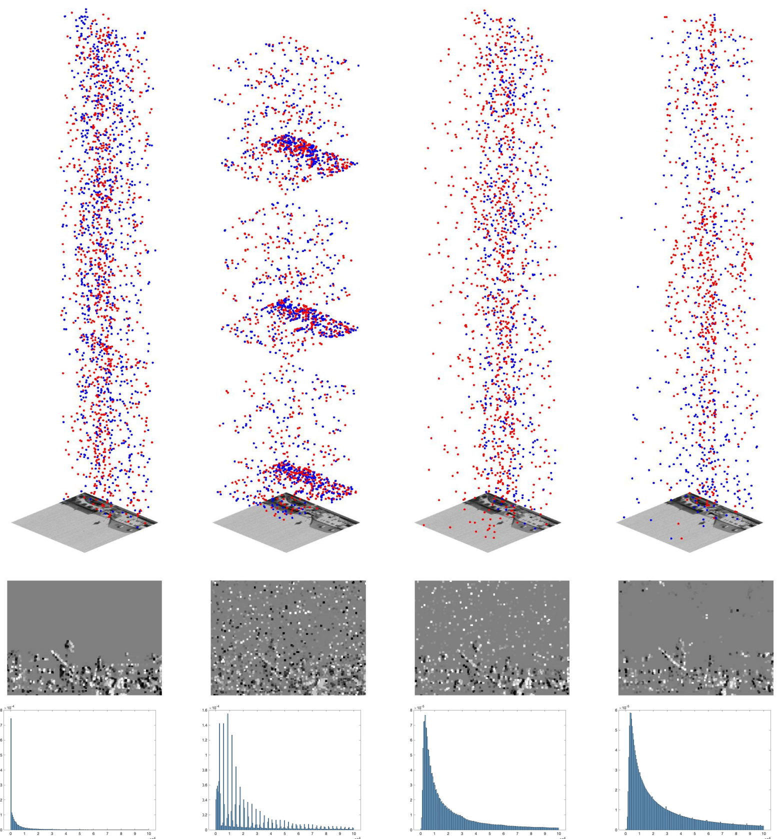


Fig. 7. Comparison on DAVIS346 'rec1487593224' data [2]. (a)(b)(c) Simulated reproduction from Vid2E [4], V2E [3], our DVS-Voltmeter. (d) real DAVIS346 data. We illustrate 3D clouds, 2D time surfaces, and probability density function histograms of event data from top to bottom. The color pair (red, blue) represents the polarity (1,-1) of events. The proposed DVS-Voltmeter gains more randomness and the generated events resemble real data.

5 More Results on Semantic Segmentation Application

In this section, we provide more visual comparison on semantic segmentation application in Fig. 8 to Fig. 10. As there is a good resemblance between real events and the simulated ones generated from our simulator, the segmentation network trained on our data can give a more accurate and detailed results.

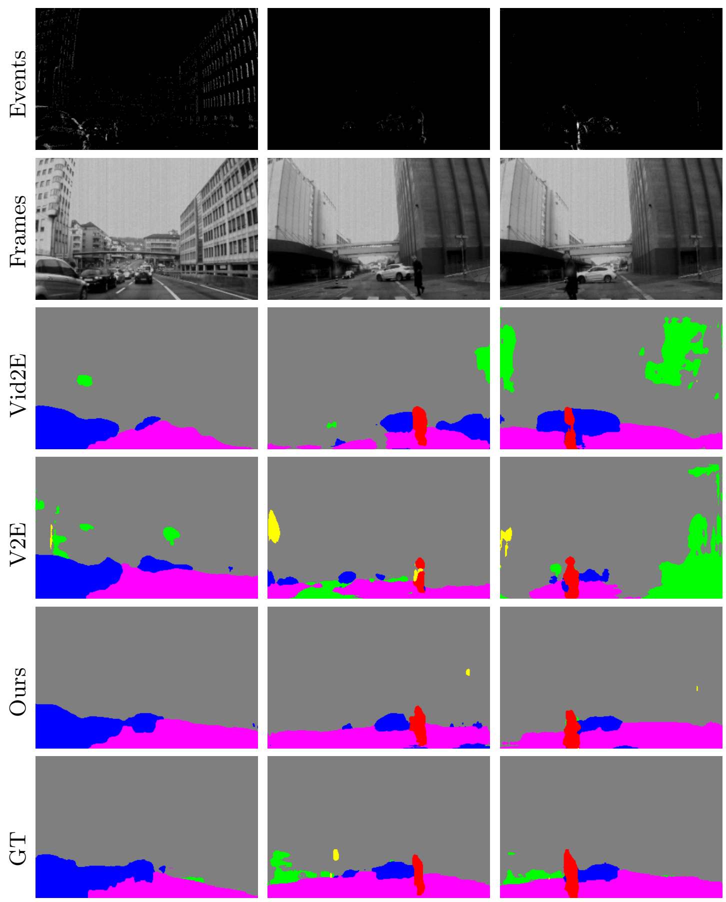


Fig. 8. Visual comparisons on semantic segmentation on Ev-Seg data [1]. The network trained on our simulated events generates much accurate and detailed results.

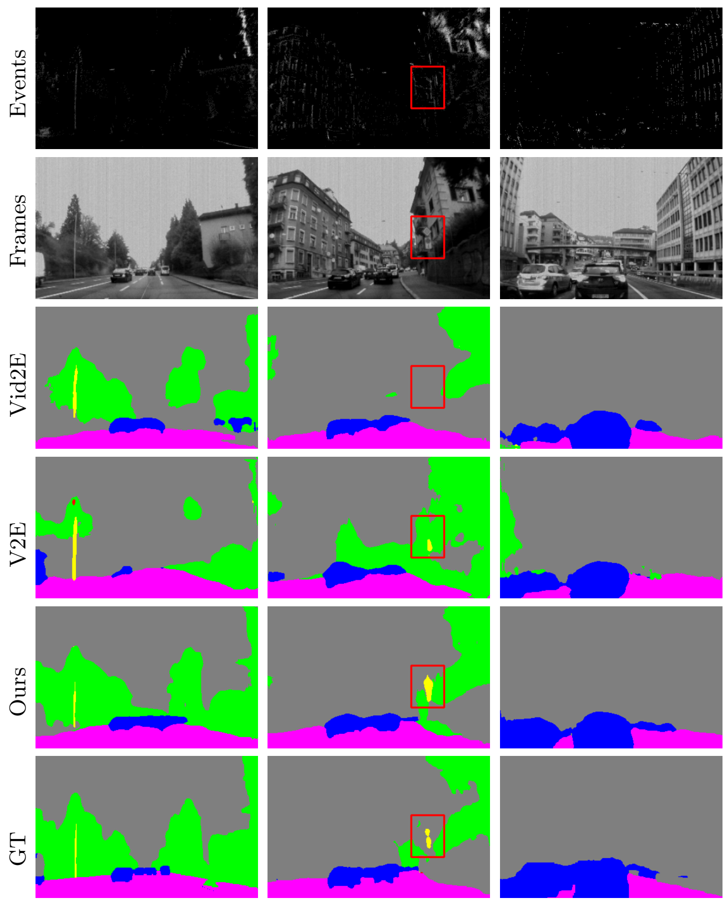


Fig. 9. Visual comparisons on semantic segmentation on Ev-Seg data [1]. The network trained on our simulated events generates much accurate and detailed results.

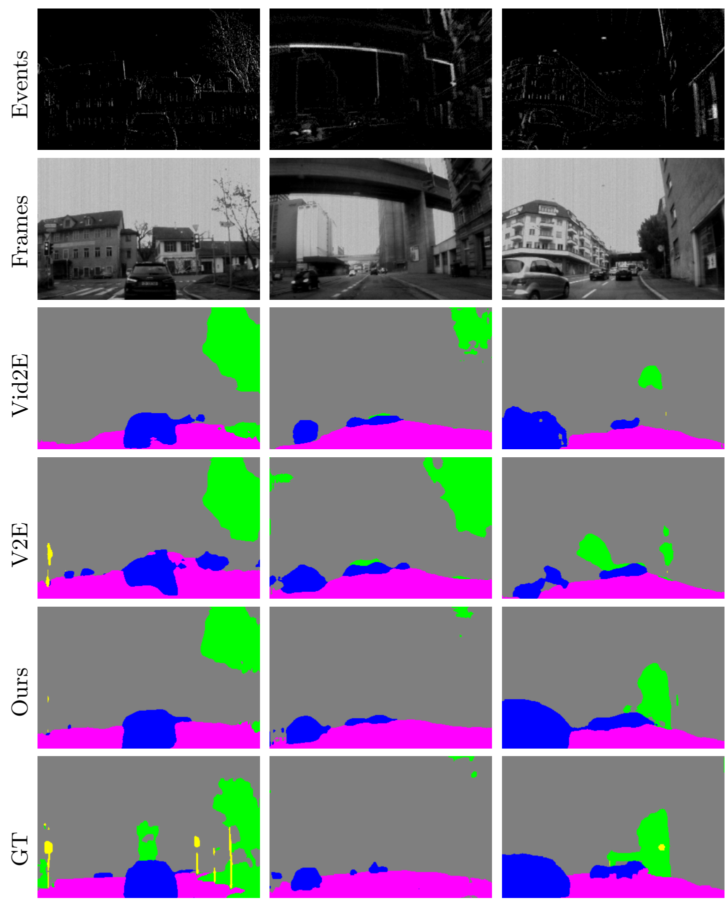


Fig. 10. Visual comparisons on semantic segmentation on Ev-Seg data [1]. The network trained on our simulated events generates much accurate and detailed results.

6 More Results on Intensity-Image Reconstruction Application

We provide more visual comparison on intensity-image reconstruction in Fig. 11 to Fig. 13. As the proposed simulator is designed based on the statistics and circuit principle of events, it naturally encourages the reconstructed images to have natural image statistics. The results show that the network trained on our simulated events reconstructs more visually pleasing images with finer details and fewer artifacts.

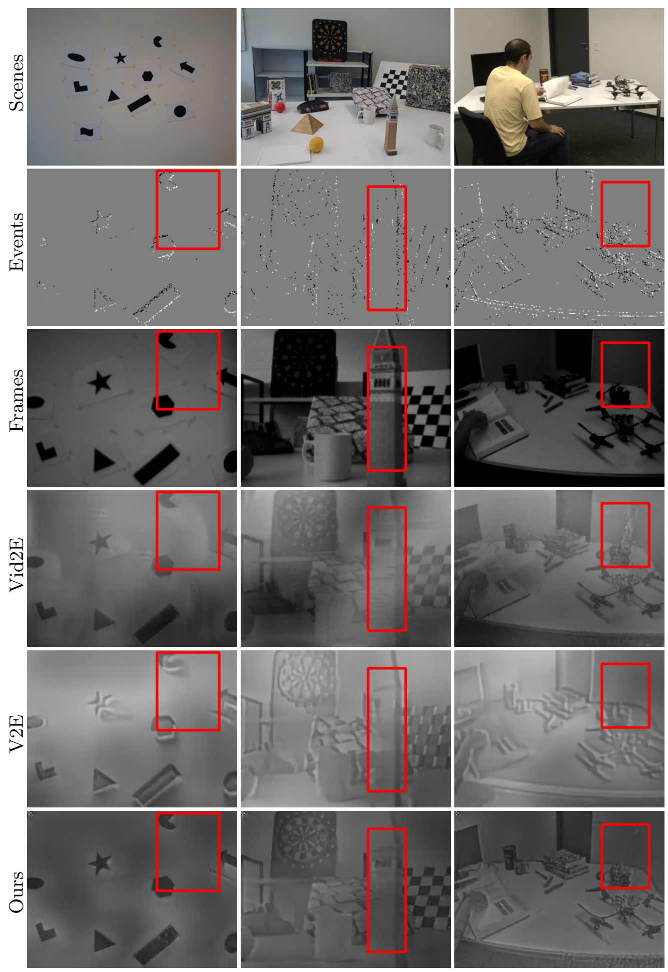


Fig. 11. Visual comparisons on intensity-image reconstruction on Event Camera Dataset [6]. The network trained on our simulated events generates much sharper results with fewer artifacts.

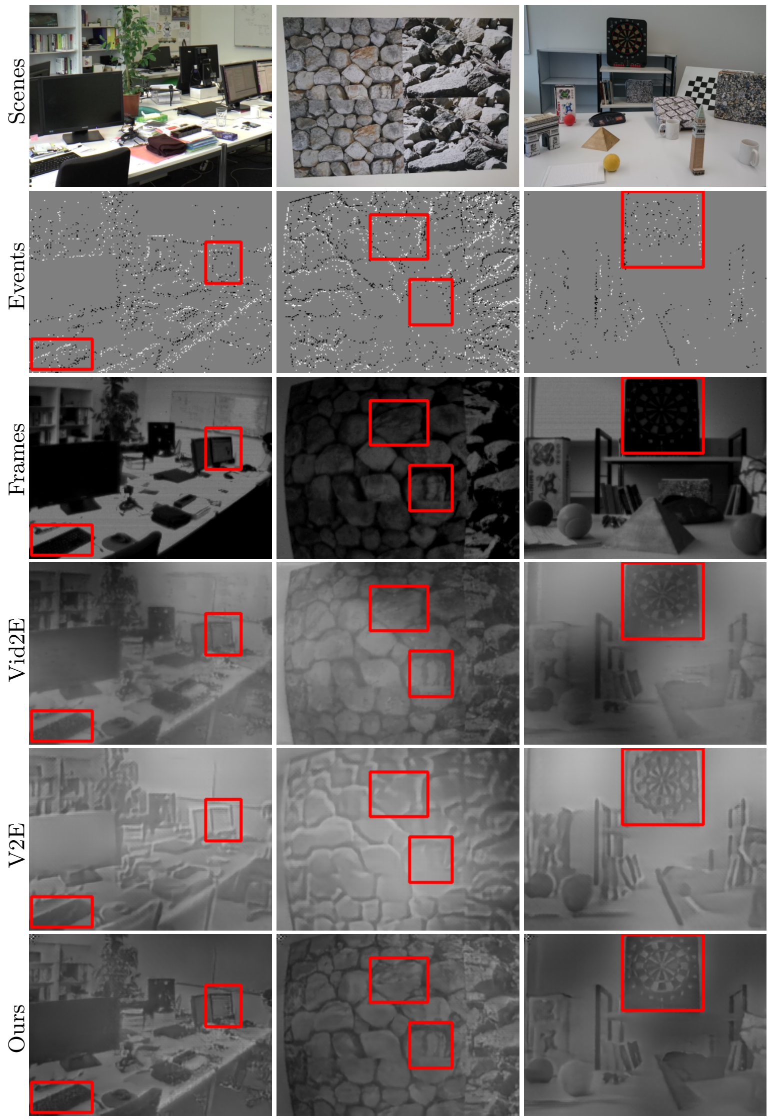


Fig. 12. Visual comparisons on intensity-image reconstruction on Event Camera Dataset [6]. The network trained on our simulated events generates much sharper results with fewer artifacts.

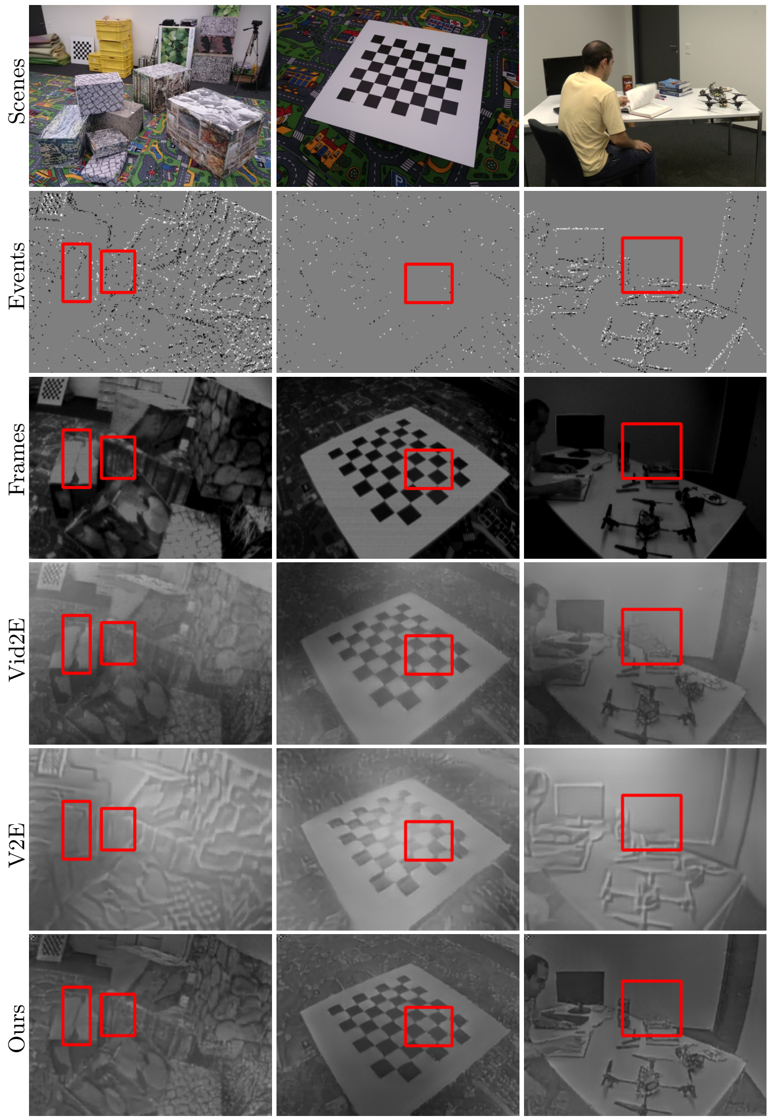


Fig. 13. Visual comparisons on intensity-image reconstruction on Event Camera Dataset [6]. The network trained on our simulated events generates much sharper results with fewer artifacts.

References

1. Alonso, I., Murillo, A.C.: Ev-segnet: Semantic segmentation for event-based cameras. In: IEEE Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 0–0 (2019)
2. Binas, J., Neil, D., Liu, S.C., Delbruck, T.: Ddd17: End-to-end davis driving dataset (2017)
3. Delbruck, T., Hu, Y., He, Z.: V2e: From video frames to realistic dvs event camera streams. arXiv preprint arXiv:2006.07722
4. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to events: Recycling video datasets for event cameras. In: IEEE Computer Vision and Pattern Recognition (CVPR). pp. 3586–3595 (2020)
5. Lagorce, X., Orchard, G., Galluppi, F., Shi, B.E., Benosman, R.B.: Hots: a hierarchy of event-based time-surfaces for pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **39**(7), 1346–1359 (2016)
6. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. The International Journal of Robotics Research (IJRR) **36**(2), 142–149 (2017)
7. Nozaki, Y., Delbruck, T.: Temperature and parasitic photocurrent effects in dynamic vision sensors. IEEE Transactions on Electron Devices **64**(8), 3239–3245 (2017)