CONFIDENTIAL EXTENDED ABSTRACT.
DO NOT DISTRIBUTE ANYWHERE.

Arbitrary Viewpoint Event Camera Simulation by Neural Radiance Fields

Diego Hernández Rodríguez[1,2,a)] Motoharu Sonogashira[2,1,b)]
Kazuya Kitano[1,c)] Yuki Fujimura[1,d)] Takuya Funatomi[1,e)]
Yasuhiro Mukaigawa[1,f)] Yasutomo Kawanishi[2,1,g)]

Abstract

Event cameras are novel sensors that offer significant advantages over standard cameras, such as high temporal resolution, high dynamic range, and low latency. Despite recent efforts, event cameras remain relatively expensive and difficult to obtain. Simulators for these sensors are crucial for developing new algorithms and mitigating accessibility issues. However, existing simulators that tackle realistic scenes often fail at generalizing to novel viewpoints or temporal resolutions, making the generation of realistic event data from a single scene not feasible. To address these challenges, we propose leveraging neural radiance fields (NeRFs) to enhance event camera simulators. NeRFs are capable of synthesizing novel views of complex scenes from a low framerate video sequence, providing a powerful tool for simulating event cameras from arbitrary viewpoints. This approach not only simplifies the simulation process but also allows for greater flexibility and realism in generating event camera data, making the technology more accessible to researchers and developers. We show that our simulator is able to approximate an event camera data stream.

1. Introduction

Event cameras represent a paradigm shift in visual sensing technology, capturing dynamic scenes with remarkable temporal resolution and high dynamic range. Unlike conventional frame-based cameras, event cameras asynchronously record changes in the intensity of the visual field, offering a unique advantage in scenarios involving fast motion or challenging lighting conditions. Since these sensors are still relatively expensive and difficult to obtain, various efforts have been made to create simulators to further facilitate their research. Previous simulators aim to generate event data from RGB video by either relying on ultra-high framerates [2, 3] or by interpolation of the video sequence [4]. This comes with the drawback of not being able to generate more data from a single video. While simulators like ESIM [9] attempt to tackle this issue with the use of 3D models, generating data that resembles a realistic scene is both time and labor-intensive, making it unsuitable for researchers who want to generate their own data. To address this challenge, we propose a framework of a simulation shown in Fig. 1. The framework generates synthetic event camera data using Neural radiance fields (NeRFs) [7], a recent breakthrough in the field of computer vision that enables the reconstruction of high-fidelity 3D scenes from a sparse set of 2D images by leveraging neural networks to model the volumetric radiance field. By integrating NeRF with event-based sensing principles, we aim to create a versatile framework that can produce realistic and diverse event camera data, facilitating the advancement of event-based vision algorithms.

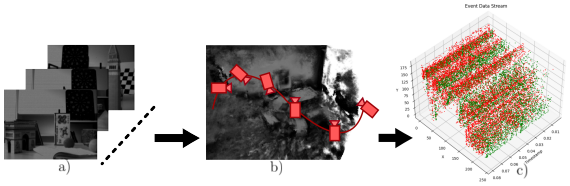


Fig. 1: Process flow of our method. A series of RGB images a) generate a radiance field b) which is sampled along a camera path to generate an event stream c)

Our approach offers several significant advantages. First, it allows for the creation of extensive datasets without the need for labor-intensive data collection processes. Second, it provides a controlled environment where various parameters can be modified to evaluate the robustness of event-based algorithms. Finally, the synthetic data generated through our method can serve as a valuable resource for training deep learning models, potentially improving their performance in real-world applications.

In section 2, we introduce some of the most important works concerning event camera simulation, explain their working mechanism, as well as doing a quick review of the formulation of neural radiance fields. In section 3, we detail

[1] Nara Institute of Science and Technology
[2] RIKEN
[a)] hernandez_rodriguez.diego.hc7@is.naist.jp
[b)] motoharu.sonogashira@riken.jp
[c)] kitano.kazuya@is.naist.jp
[d)] fujimura.yuki@is.naist.jp
[e)] funatomi@is.naist.jp
[f)] mukaiwaga@is.naist.jp
[g)] yasutomo.kawanishi@riken.jp

the methodology for synthesizing event camera data using NeRF, discuss the implementation and integration of these technologies, and present experimental results demonstrating the effectiveness of our approach. By bridging the gap between synthetic data generation and event-based sensing, our work aims to accelerate research in the field of event cameras, paving the way for their broader adoption and application. In section 4, we discuss our results, comparing them to actual event data streams and with other video-to-event generation pipelines. Finally, in section 5, we discuss the limitations of our method, as well as possible extensions and future work.

2. Related Work

In recent years, event camera datasets and simulators have been introduced. In this section, we briefly review the most important ones and their specific application scenarios. We also do a quick review of the mechanism behind neural radiance fields. We then turn to the work on the simulation of an event camera.

2.1 Event camera simulation

The number of event camera simulators publicly released is small, and while some of them build upon previous research, they mostly tackle the task in different manners. Some simulators [4, 5] attempt to model the unique characteristics of the sensor and its parameters. However, none of the simulators described take into account the geometry of the scene, nor can they generate an event stream outside of the original path followed by the camera. ESIM [9] (part of the Vid2E pipeline [4], leverages a deep learning-based approach in order to upsample a video stream and generate a continuous event stream from sparse images.

2.2 Neural radiance fields

Neural radiance fields represent a scene utilizing a multilayer perceptron (MLP) $F_\theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ that maps a position in 3D space $\mathbf{x} = (x, y, z)$ and a 2D viewing direction $\mathbf{d} = (\theta, \phi)$ to its corresponding directional emitted radiance, in other words, its color $\mathbf{c} = (R, G, B)$ and volume density σ. From this representation, the estimated emitted radiance $\widehat{\mathbf{L}}$ at a given pixel $\mathbf{u}$ can be calculated using the volume rendering equation [11] with quadrature, as follows:

$$\widehat{\mathbf{L}}(\mathbf{u}) = \sum_{k=1}^{N} T_k (1 - \exp(-\sigma_k \delta_k)) \mathbf{c}_k, \quad \text{where} \quad T_k = \exp(-\sum_{m=1}^{k-1} \sigma_m \delta_m), \tag{1}$$

where σ_k and $\mathbf{c}_k$ are the volume density and the emitted radiance, respectively, of a sampled position $\mathbf{x}_k$ along the back-projected ray $\mathbf{r}$ through a pixel, which has a direction $\mathbf{d}$ and an origin $\mathbf{o}$ at the camera center. The sample $\mathbf{x}_k = \mathbf{o} + s_i \mathbf{d}$ has a distance s_k from the camera center and a distance of $\delta_k = s_{k+1} - s_k$ between its adjacent sample $\mathbf{x}_{k+1}$.

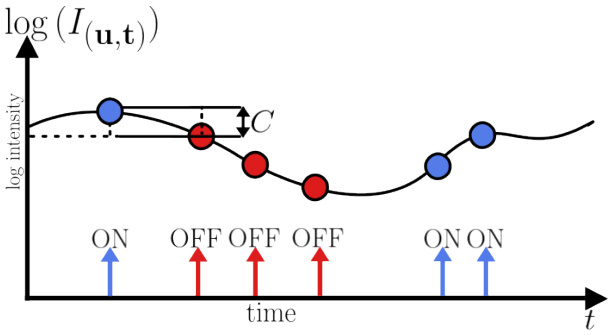


Fig. 2: A pixel $\mathbf{u}$ of the intensity image I_t in the event generation model. A positive or negative event is generated when the brightness change exceeds the threshold C in a logarithmic scale. Represented in blue and red, respectively.

3. Method

3.1 Problem formulation

Following the convention of ESIM [9] event camera data is represented as $\mathbf{e}_k = (t_k, \mathbf{u}_k, p_k)$, denoting brightness changes asynchronously registered by a pixel at time t_k, pixel location $\mathbf{u}_k = (x_k, y_k)$ in the camera frame, with a polarity $p_k \in \{-1, 1\}$. The polarity of an event indicates a positive or negative change in illumination according to a logarithmic scale, quantized by negative and positive thresholds $\pm C$. The change in brightness between two timestamps can be estimated by the difference of intensity images I_t and I_{t-1} in the logarithmic scale. This mechanism is illustrated in Fig. 2.

$$\Delta \log(I) = \log(I_t) - \log(I_{t-1}), \tag{2}$$

$$\mathbf{e}_{p_k} = \begin{cases} -1 & \text{if} \quad C \leq \Delta \log(I), \\ 1 & \text{if} \quad C \geq \Delta \log(I). \end{cases} \tag{3}$$

3.2 Event data generation by sampling radiance fields

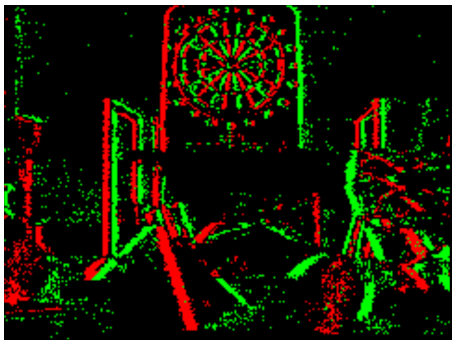
We aim to generate a simulated event camera data stream from a sequence of RGB images. We first train a radiance field $\mathbf{F}$ on that sequence. Following the methodology behind ESIM's event generation from 3D models, our method approximates the per pixel value of the intensity image $\Delta \log(I)$ by sampling a camera path along $\mathbf{F}$, and calculating the color of each pixel by accumulating the contributions from all sampled points along the ray following equation (1). Since event cameras operate in brightness pixels, we convert the sampled color images using the ITU-R Recommendation BT.601 for luma, i.e., according to the formula:

$$Y = 0.299R + 0.587G + 0.114B, \tag{4}$$

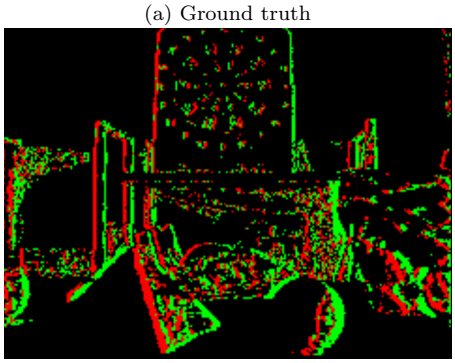
with RGB channels in linear color space. This yields the following equation:

$$I_{(\mathbf{u})} = Y(\widehat{\mathbf{L}}(\mathbf{u})). \tag{5}$$

Generating a pair of logarithmic intensity images $\log(I_t)$ and $\log(I_{t-1})$ based on user-defined parameters, such as



(a) Ground truth



(b) Ours

Fig. 3: Comparison of event streams, positive and negative events are colored red and green, respectively.

maximum number of events per camera position, pixel refraction period, and brightness change threshold C. The exact threshold of an actual event camera is not known, so we decided to set $C = 0.25$ for our experiments.

We can then determine the number of predicted events at a certain pixel location during that time window with the following equation:

$$\mathbf{e_u} = \begin{cases} \dfrac{\Delta \log(I_{(\mathbf{u})})}{+C} & \text{if} \quad \Delta \log(I_{(\mathbf{u})}) \geq 0, \\ \dfrac{\Delta \log(I_{(\mathbf{u})})}{-C} & \text{if} \quad \Delta \log(I_{(\mathbf{u})}) \leq 0. \end{cases} \tag{6}$$

4. Experiments

We conduct our experiments on the dataset provided by Mueggler et al. [8] for our comparisons since it contains images generated by a DAVIS sensor [1] which are used to train the radiance field, as well as camera positions from an external tracker, eliminating the need to use COLMAP [10] for camera pose estimation.

4.1 Experimental results

4.1.1 Comparison with real event camera data

In order to perform our tests, we interpolate five equidistant positions between each camera pose along the initial camera path, akin to the frame interpolation V2E does. The result of our simulation can be observed in Fig. 3.

Due to the inherent challenge in accurately modeling the noise characteristics of an event camera sensor, as well as the randomness it introduces into the firing pixels, we have chosen to simulate its ideal operation instead. However, it is possible to set a noise parameter, as well as a hot pixel parameter.

Table 1: Comparison of PSNR (dB) values obtained in scenes from the dataset [8] (higher is better).

Scene name	Ours	V2E [4]
slider	**30.01**	29.40
boxes 6dof	**28.32**	28.06
poster	28.04	**28.57**

4.1.2 PSNR of accumulated event frames

In order to measure the correctness of the simulated events, we perform an accumulation operation on both the ground truth and simulated event streams to generate a frame representation. The accumulation operation integrates events over time into a frame-by-frame basis, aggregating changes captured by the sensor. As shown in [8], a logarithmic intensity image $\log \hat{I}(\mathbf{u}; t)$ can be reconstructed from the event stream at any point in time t by accumulating events:

$$\begin{aligned} \log \hat{I}(\mathbf{u}; t) &= \log I(\mathbf{u}; 0) + \gamma \delta(t - t_k), \\ \text{where} \quad \gamma &= \sum_{0 < t_k \leq t} p_k C \delta(\mathbf{u} - \mathbf{u}_k), \end{aligned} \tag{7}$$

We utilize a modified version of this functions which applies a decay parameter to reduce the noise of the generated frame. The accumulator function applies an exponential decay $d(t, \tau)$ to equation (7):

$$\begin{aligned} \log \hat{I}(\mathbf{u}; t) = &\log \big(I(\mathbf{u}; 0) d(t, \tau) + I(\mathbf{u}; n)(1 - d(t, \tau)) \\ &+ \gamma d(t - t_k, \tau) \big), \\ \text{where} \quad d(t, \tau) = &\exp\left(-\frac{t}{\tau} \right), \end{aligned} \tag{8}$$

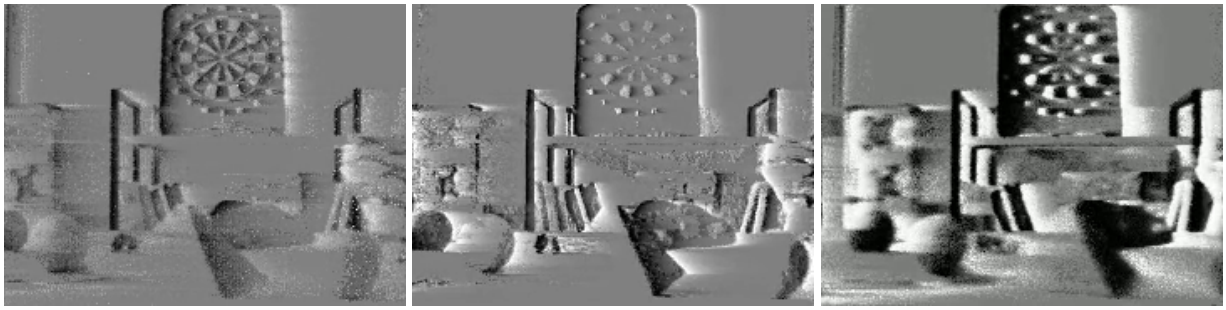
where $\log(I(\mathbf{u}; 0))$ is the logarithm of the intensity of the pixel at the previous accumulated frame, $\log(I(\mathbf{u}; n))$ is a neutral potential and the decay parameter is the time constant τ. For our experiments we set $\tau = 1 \times 10^{-5}$ microseconds and $\log(I(\mathbf{u}; n)) = 0.5$.

After accumulating all the events and generating a video sequence, we calculate the peak signal-to-noise ratio between the ground truth and the simulated event stream. We also utilize V2E as a baseline for video-to-event simulation. The results of this experiment can be seen in Fig. 4.

5. Limitations and extensions

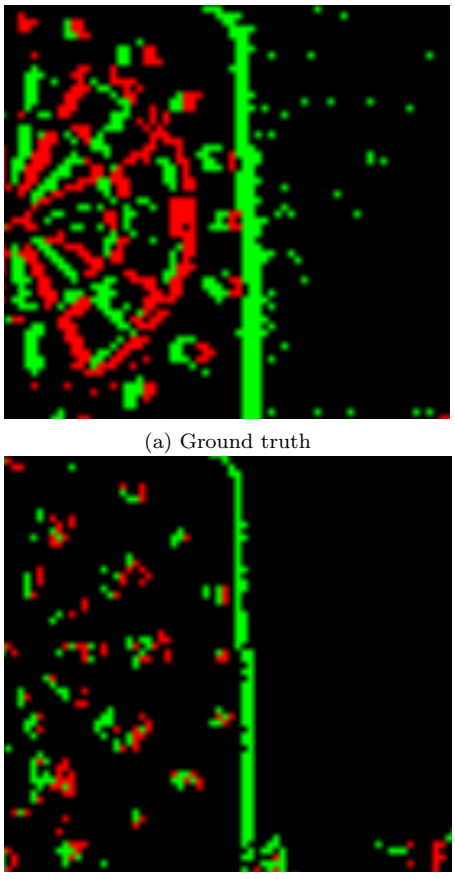
As demonstrated in Fig. 3, our simulator correctly approximates the positive and negative events measured by an actual event camera. It is worth noting that due to not including both noise and hot pixel simulation in our experiments, some areas of the simulation appear to not show any information registered; a zoom-in of an extreme case is illustrated in Fig. 5

While this paper primarily focuses on the application of radiance fields for static scene reconstruction, it is important to note several limitations and potential avenues for future research.



(a) Ground truth (b) Ours (c) V2E [4]

Fig. 4: Visual comparison of accumulated frames



(a) Ground truth

(b) Ours

Fig. 5: Comparison of event streams, positive and negative events are colored red and green, respectively.

5.1 Dynamic scene reconstruction

Radiance fields have the ability to reconstruct dynamic scenes, the NeRF backbone utilized did not have the capabilities to represent dynamic scenes, so we leave their implementation as a task for future research.

5.2 Integration with different representations

Our simulator, by its design, does not rely on a specific representation of radiance fields. This flexibility allows for easy integration with alternative rendering techniques such as Gaussian splatting [6].

5.3 Generalization to Real-world Applications

While our simulator demonstrates promising results in controlled environments, generalizing these findings to real-world applications presents additional challenges. Factors such as varying lighting conditions, occlusions, and reflective surfaces can significantly impact the performance and accuracy of radiance field reconstruction.

6. Conclusion

In this paper, we introduced a novel method for event camera simulation using neural radiance fields. Our approach leverages the capabilities of NeRFs to synthesize novel views of complex scenes, enabling the generation of realistic and diverse event camera data from arbitrary viewpoints. Experimental results demonstrate that our simulator matches or outperforms existing methods in terms of accuracy and realism, providing a valuable tool for the development and evaluation of event-based vision algorithms. The key contributions of this work include the integration of NeRFs with event-based sensing principles and the development of a versatile and efficient event camera simulator. We believe that this method represents a significant advancement in the field of event camera simulation, making this technology more accessible to researchers and developers.

References

[1] Brandli, C., Berner, R., Yang, M., Liu, S.-C. and Delbruck, T.: A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor, *IEEE Journal of Solid-State Circuits*, Vol. 49, No. 10, pp. 2333–2341 (2014).

[2] García, G. P., Camilleri, P., Liu, Q. and Furber, S.: pyDVS: An extensible, real-time Dynamic Vision Sensor emulator using off-the-shelf hardware, *IEEE Symposium Series on Computational Intelligence*, pp. 1–7 (2016).

[3] Gehrig, D., Gehrig, M., Hidalgo-Carrió, J. and Scaramuzza, D.: Video to Events: Recycling Video Datasets for Event Cameras, *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020).

[4] Hu, Y., Liu, S. C. and Delbruck, T.: v2e: From Video Frames to Realistic DVS Events, *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE (2021).

[5] Joubert, D., Marcireau, A., Ralph, N., Jolley, A., van Schaik, A. and Cohen, G.: Event Camera Simulator Improvements via Characterized Parameters, *Frontiers in Neuroscience*, Vol. 15 (2021).

[6] Kerbl, B., Kopanas, G., Leimkühler, T. and Drettakis, G.: 3D Gaussian Splatting for Real-Time Radiance Field Rendering, *ACM Transactions on Graphics*, Vol. 42, No. 4 (2023).

[7] Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R. and Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, *ECCV* (2020).
[8] Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T. and D.: Scaramuzza: The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM, *International Journal of Robotics Research*, Vol. 36, pp. 142–149 (2017).
[9] Rebecq, H., Gehrig, D. and Scaramuzza, D.: ESIM: an Open Event Camera Simulator, *Conference on Robot Learning (CoRL)* (2018).
[10] Schönberger, J. L. and Frahm, J.-M.: Structure-from-Motion Revisited, *IEEE Conference on Computer Vision and Pattern Recognition* (2016).
[11] Tagliasacchi, A. and Mildenhall, B.: Volume Rendering Digest (for NeRF), *arXiv:2209. 02417 [cs]*, Vol. 3, p. 02417 (2022).