Ev3DGS: Event Enhanced 3D Gaussian Splatting from Blurry Images

Junwu Huang, Zhexiong Wan, Zhicheng Lu, Juanjuan Zhu, Mingyi He, Yuchao Dai†
School of Electronics and Information & Shaanxi Key Laboratory of Information Acquisition and Processing
Northwestern Polytechnical University, Xi'an, Shaanxi 710129, China

***Abstract*—The novel view synthesis task involves inputting a source image, a source pose and a target pose, and rendering to generate a corresponding target image. However, obtaining a clear novel view synthesized image from only a set of blurred images and corresponding poses is a challenging problem, which often occurs in reality. To solve this problem, the good performance of 3D Gaussian Splatting (3DGS) in the field of 3D scene reconstruction is taken into account, as well as the remarkable effectiveness of event cameras in the deblurring problem. Inspired by the novel Event-Enhanced Neural Radiance Fields (E2NeRF) model, which is also based on event enhancement, a new 3D reconstruction framework, Event-Enhanced 3DGS (Ev3DGS), based on 3DGS is proposed by utilizing the combined data from event cameras and standard RGB cameras. We effectively introduce the event stream into the 3D Gaussian iterative process by constructing the blur rendering loss and event rendering loss, which guides the optimization of the network structure by predicting the blurred image and event generation processes. Compared with E2NeRF, the Ev3DGS proposed in this paper effectively improves the rendering performance with 4.8% and 2.5% improvement in PSNR and SSIM, and 15% reduction in LPIPS, while significantly reducing the training time consumption. Extensive experiments on both synthetic and real-world datasets show that Ev3DGS can effectively learn clear 3DGS from blurred image inputs, enabling high-quality novel view synthesis and making 3DGS more robust. Our code is publicly available at https://github.com/npucvr/Ev3DGS.**

I. Introduction

The novel view synthesis has a wide range of applications in 3D scene reconstruction, virtual/reality augmentation and other fields. In recent years, with the development of differentiable rendering technology, the novel view synthesis has received widespread attention again, and has shown superior performance and development prospects. Among them, the appearance of 3D Gaussian Splatting (3DGS) [1] has greatly accelerated the rendering speed of novel view synthesis. However, it is difficult for 3DGS to obtain a clear target image from a blurred source image input, as often occurs in reality, which poses a challenge to synthesize clear novel views using 3DGS.

Event camera, as a new type of bionic vision sensor, has a wide range of applications in the fields of feature detection and tracking, optical flow estimation [2] [3], video frame prediction [4], 3D reconstruction and pose estimation. It is promising to guide the learning process of 3DGS by introducing additional information carried by event data. Therefore, in order to solve the problem of 3DGS input source image blurring, this paper investigates event-enhanced 3DGS for blurred images.By utilizing event and image data acquired by an event camera and introducing blur rendering loss and event rendering loss, a new event-enhanced 3DGS-based model (Ev3DGS) is constructed, which enables Ev3DGS to efficiently learn from blurred images to a clear Gaussian scenes, thus realizing clear novel view synthesis.

Specifically, inspired by the Event-Enhanced Neural Radiance Fields (E2NeRF) [5] model, during the training process, we superimpose the clear images obtained by rendering multiple predicted poses at equal time intervals under one viewpoint as the predicted blurred images, and compare them with the input blurred images as our blur rendering loss. Secondly, the generation process of predicted event data is simulated based on the brightness change caused by the change of camera position and compared with the real event data to get the event rendering loss. By taking advantage of the high dynamic range, high temporal resolution, and low latency of the event camera, the event data can effectively enhance the 3DGS network so that we can learn a network with a clearer output image, which makes it possible to not only realize the deblurring of the input image, but also obtain high-quality synthetic images of the novel view.

Our proposed Event-Enhanced 3DGS (Ev3DGS) in this paper is tested and verified with E2NeRF synthetic dataset and real-world dataset. The comparisons are carried out with the E2NeRF model to test the effectiveness of the model. In addition this paper also carries out ablation experiments to demonstrate the effectiveness of event rendering loss. The contributions of this paper can be summarized as follows:

(1) An event-enhanced 3DGS model (Ev3DGS) is proposed, which is the first framework to reconstruct clear 3DGS from blurred images and corresponding event data using blur rendering loss and event rendering loss. The modeling framework effectively exploits the intrinsic relationship between events and images to significantly improve the performance and robustness of 3DGS;

(2) Both the Blur-rendering loss and event-rendering loss are introduced into 3DGS, which enhances the object clarity of reconstruction from 3DGS.

† Corresponding author (daiyuchao@nwpu.edu.cn)

This research was partly supported by the National Natural Science Foundation of China (62271410) and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX2023013).

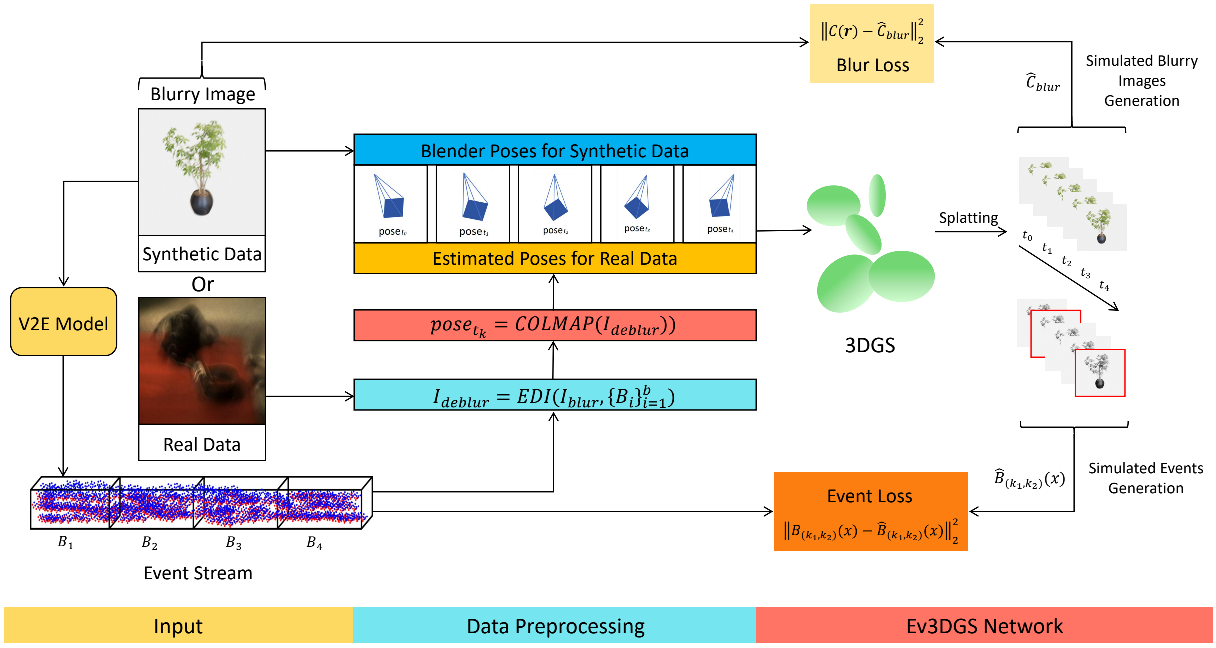

Fig. 1. Ev3DGS Model Framework. Its inputs are blurry images and corresponding event streams of every views.This Ev3DGS model introduces a blur rendering loss and an event rendering loss. The blur rendering loss simulates the generation process of blurred images to provide more information about the scene texture details to the network. The event rendering loss, on the other hand, introduces event data into the 3DGS training process, enabling the network to better learn the real 3D volume representation and obtain information about the motion characteristics of objects.

II. Related Work

A. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [1] is a transformative technology in computer graphics in recent years. Recent innovative researches employing 3DGS in SLAM (Simultaneous Localization and Mapping) have demonstrated the great potential and versatility of 3DGS in SLAM. For example, GS-SLAM [6] employs an adaptive strategy to add or remove 3D Gaussian distributions to optimize the scene geometry reconstruction and improve the mapping of previously observed regions. To extend the concept of 3DGS to dynamic scenes, Lu *et al.* [7] proposes a 3D geometry-aware deformable Gaussian Splatting method for dynamic view synthesis, while SC-GS [8] utilizes sparse control points and deformed Multilayer Perceptrons (MLPs) to capture and represent the dynamics of a 3D scene. For the problem of blurring of the input image due to camera shake, Deblur-GS [9] relied on the interpolation method to estimate the motion trajectory information of the camera to realize the reconstruction of a clear 3D scene. BAD-Gaussian [10] performs batch clustering on top of this, which enhances the interrelationships of the images in the same exposure time and can better capture the complex motion trajectories during the exposure period.

B. Event Camera

Event Camera is a new type of biologically inspired vision sensor, sometimes called Dynamic Vision Sensor (DVS, dynamic vision sensor) or DAVIS (Dynamic and Active-Pixel Vision Sensor) [11]. Unlike standard cameras that capture images at a fixed frame rate, event cameras generate event streams by sensing changes in pixel brightness, which is characterized by low latency, high dynamic range, low power consumption, and high temporal resolution. It is also widely used in some traditional vision tasks such as feature detection and tracking, optical flow estimation, 3D reconstruction and pose estimation. In the field of optical flow estimation, Zhu *et al.* [12] utilized a self-supervised scheme to train the encoder and decoder of Convolutional Neural Network (CNN) for dense optical flow estimation, while Ye *et al.* [13] proposed an innovative monocular neural network structure that relies only on event data to estimate dense optical flow, depth and self-motion. In the field of 3D reconstruction, many active event-based 3D reconstruction methods have also been proposed. For example, Brandli *et al.* [14] successfully combined a DVS with a pulsed line laser for fast terrain reconstruction. Motion-contrast 3D scanning [15], on the other hand, uses structured light technology to synchronize high resolution, high speed and excellent performance in complex 3D scanning environments. For the event-enhanced new perspective synthesis task, Qi *et al.* [5] proposed the Event-Enhanced Neural Radiance Fields model E2NeRF, which constructs two losses and well utilizes the event properties to assist in solving the problem of blurring of the input image and realizes a clear three-dimensional reconstruction, but its training speed is slow due to the limitation of the Neural Radiance Fields (NeRF) model.

III. Prerequisite

A. 3D Gaussian Splatting

3D Gaussian Splatting [1] is an explicit rasterization technique for real-time radial field rendering described by 3D

Gaussian distributions, which allows for real-time rendering of photorealistic scenes learned from small image samples. The input to 3D Gaussian Splatting is a set of images of a static scene and a sparse point cloud obtained by the camera from the alignment. On the sparse points, a set of 3D Gaussian distributions are created, defined by the position $\boldsymbol{x}$, opacity α, covariance matrix $\boldsymbol{\Sigma}$, and Spherical Harmonic (SH) coefficients to model the view-dependent color, while parameter optimization is performed by an adaptive density control algorithm. The Gaussian distribution is defined as:

$$G(\boldsymbol{x}) = e^{-\frac{1}{2}(\boldsymbol{x})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x})}. \tag{1}$$

For regular optimization of the covariance matrix $\boldsymbol{\Sigma}$, given a scaling matrix $\boldsymbol{S}$ and a rotation matrix $\boldsymbol{R}$, the corresponding $\boldsymbol{\Sigma}$ can be found:

$$\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}. \tag{2}$$

To allow independent optimization of these two factors, they are further represented as a three-dimensional vector $\boldsymbol{s}$ for scaling and a quaternion $\boldsymbol{r}$ for rotation.

In addition, the key to 3DGS's improved model rendering rate is the construction of a fast micronizable rasterizer. The overall fast rendering and sorting is achieved by chunking the Gaussian sphere splash and allowing the Gaussian spheres with approximate a-values to be blended. And the fast rasterizer utilizes fast inverse passes to track through the accumulated a-values, allowing it to receive splat balls with gradients without restriction for scene representation. For each pixel, its color C is determined by all Gaussian distributions covering the pixel, which are represented as follows:

$$C = \sum_{i \in N_{cov}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{3}$$

where N_{cov} represents the splats that cover this pixel, α_i represents the opacity of this Gaussian splat multiplied by the density of the projected 2D Gaussian distribution at the location of the pixel, and c_i represents the computed color.

B. Event Generation

The event camera is a novel bionic vision sensor that asynchronously and independently measures the brightness change of each pixel in the scene, and triggers the generation of an event signal when the brightness change of any pixel exceeds a threshold set by the event camera. Each event consists of the spatio-temporal coordinates of the triggering event (pixel position coordinates (x, y) and the triggering timestamp t with millisecond accuracy) and its polarity $\sigma = \pm 1$. The event camera is able to measure the brightness change of each pixel in the scene asynchronously and independently [16].

However, when the object moves, the event camera asynchronously generates event signals by sensing changes in image intensity, producing a series of event sequences, denoted as:

$$\sigma = \boldsymbol{\Gamma}\left(\log\left(\frac{L_{xy}(t)}{L_{xy}(t_{ref})}\right), c\right), \tag{4}$$

where the potential image $L_{xy}(t)$ and $L_{xy}(t_{ref})$ represents the intensity at point (x, y) at moment t and t_{ref}, c is the response threshold for intensity change, and $\boldsymbol{\Gamma}(\cdot,\cdot)$ is a truncated function:

$$\boldsymbol{\Gamma}(d, c) = \begin{cases} +1, & d \geq c \\ 0, & d \in (-c, c) \\ -1, & d \leq -c \end{cases}, \tag{5}$$

where d is the intensity of the corresponding brightness change.

According to the general principles of event generation, the most primitive and complete representation of all the information in each event is to represent the event as an n × 4 dimensional matrix, where each column contains all the information of a single event: the 2D coordinates, the timestamp and the event polarity. Since only the brightness change of the object due to the motion process is of interest, in this paper, we consider equating the event data into b blocks and discretizing the time to superimpose the event data between any two potentially clear images to obtain the event change data containing only the 2D coordinates and event polarity.

C. Blurry Image Generation

The specific process of generating an image by means of a color camera can be represented as the process by which the camera sensor collects photons during the exposure and converts them into measurable charges. The degree of motion blur in the image depends on the motion of the camera during the exposure time. For example, a slow moving camera produces minimal relative motion, especially at shorter exposure times, whereas a fast moving camera produces a motion blurred image, especially in low light scenes with longer exposure times. The phenomenon may be expressed as an integral over a series of virtual potentially sharp images, denoted as follows:

$$I_{blur} = \phi \int_0^{\tau} I_t dt, \tag{6}$$

where I_{blur} denotes the real captured motion blurred image, ϕ is used as a normalization factor, τ is the camera exposure time, and I_t is the potentially clear image captured at timestamp $t \in [0, \tau]$ during the exposure time. The blurred image I_{blur} due to camera motion during the exposure time is calculated by averaging the potentially clear image I_t for each different timestamp t. The model is based on a discrete approximation. The discrete approximation of the model depends on the number k of discrete moments, denoted as follows:

$$I_{blur} = \frac{1}{b+1} \sum_{k=0}^{b} I_k. \tag{7}$$

Thus, for the blurred input of 3DGS, we can consider it as a superposition of a series of potentially clear images. So when acquiring the real world dataset and when making the synthetic dataset, we consider the blurred image at one viewpoint as an average of $b + 1$ potentially clear images. The preprocessing for the model input data will also be centered around this.

IV. METHOD

Fig. 1 shows the overall framework of Ev3DGS, which effectively improves the volumetric representation of 3DGS by introducing two new losses in the framework of 3DGS, and also designs an event-blurred image-based bit-pose estimation framework to efficiently deal with real-world data.Ev3DGS takes the blurred image's and the corresponding events as inputs for each viewpoint. Meanwhile Ev3DGS model introduces blur rendering loss and event rendering loss. The blur rendering loss simulates the process of blurred image generation and provides more information about the scene texture details to the network. The event rendering loss, on the other hand, introduces event data into the 3DGS training process, enabling the network to better learn the real 3D volumetric representation and obtain information about the motion characteristics of objects. After inputting the image bitmap into 3DGS, 3DGS generates an initial point cloud via for reconstruction rendering. After each rendering, the predicted blurred image is obtained by averaging the image sequences cumulatively, while the predicted event stream is calculated by converting the image sequences into gray-scale maps. The input blurred image and event stream are compared with the predicted values to obtain the blur rendering loss and event rendering loss for model supervision.

A. Blur Rendering Loss

With $b{+}1$ poses $\{\boldsymbol{P}_k\}_{k=0}^{b}$ of each view, we can get $b{+}1$ rays $\{\boldsymbol{r}_k\}_{k=0}^{b}$ emitted from each pixel. With the 3DGS network, each pixel can get $b+1$ color values $\left\{\widehat{C}_k = C\left(\boldsymbol{r}_k\right)\right\}_{k=0}^{b}$ regarded as the process of blurry pixel generation, and the average of the results is used as the predicted blurry color:

$$\hat{C}_{blur} = \frac{1}{b+1}\sum_{k=0}^{b} C\left(\boldsymbol{r}_k\right). \tag{8}$$

The loss function of 3DGS with blurred image as input is then expressed as:

$$\mathcal{L}_{blur} = \sum_{\boldsymbol{r}\in\mathcal{R}}\left[\left\|\hat{C}^{c}_{blur} - C(\boldsymbol{r})\right\|_2^2 + \left\|\hat{C}^{f}_{blur} - C(\boldsymbol{r})\right\|_2^2\right], \tag{9}$$

where 3DGS is designed using joint optimization of coarse and fine models.

3DGS uses stochastic gradient descent, utilizing standard GPU acceleration frameworks and adding custom CUDA cores for some operations. A sigmoid function is used for α to keep it constrained to $[0, 1)$, and an exponential activation function is used for the scaling factor of the covariance to ensure a smooth gradient.

Initialize the covariance to an isotropic Gaussian whose axis length is the same as the mean of the distances to the 3 nearest points. Standard exponential decay scheduling techniques were used for the location of the Gaussian. The loss function is the loss $\mathcal{L}_2$ and D-SSIM terms:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_2 + \lambda\mathcal{L}_{D-SSIM}, \tag{10}$$

where λ is a weighting factor.

The loss $\mathcal{L}_2$ and D-SSIM terms are standardized metrics.The loss function for 3DGS is slightly different from that of NeRF. Due to the high computational cost of NeRF by light sampling the sample points, NeRF is usually at the pixel level, whereas 3DGS performs the computation of the loss at the image level.

B. Event Rendering Loss

Image blur generation is a continuous process, however blur rendering loss only uses discrete $b{+}1$ frames corresponding to $b{+}1$ poses in the blurred image to model the blurring process, losing a large amount of information in the blur generation process. Therefore by introducing event rendering loss, the high temporal resolution property of event data is utilized to supervise the continuous blurring process between any two predicted frames. Given a pixel $\boldsymbol{x} = (x, y)$, two color values of C_{k_1} and C_{k_2} $(k_1 < k_2)$ are randomly selected from the $\left\{\widehat{C}_k\right\}_{k=0}^{b}$ of that pixel, and they are converted to grayscale values to obtain L_{k_1}, L_{k_2}. According to the principle of event generation, the difference between the logarithmic values of the two grayscale values of L_{k_1}, L_{k_2} is divided by a threshold θ to estimate the number of events between any two frames for a given pixel $\boldsymbol{x}$:

$$\widehat{B}_{(k_1,k_2)}(\boldsymbol{x}) = \begin{cases} \left[\frac{\log\left(L_{k_2}\right)-\log\left(L_{k_1}\right)}{\theta_{neg}}\right], L_{k_2} < L_{k_1} \\ \\ \left[\frac{\log\left(L_{k_2}\right)-\log\left(L_{k_1}\right)}{\theta_{pos}}\right], L_{k_2} \geq L_{k_1} \end{cases}. \tag{11}$$

The mean square error between the estimated number of events $\widehat{B}_{(k_1,k_2)}(\boldsymbol{x})$ and the actual number of events $B_{(k_1,k_2)}(\boldsymbol{x})$ in $\{B_k\}_{k=1}^{b}$ is taken as our event rendering loss. For pixel $\boldsymbol{x}$ corresponding to $B_k(\boldsymbol{x})$ in the event block, set the number of negative events to its additive inverse so that positive and negative events can cancel each other out when stacking event blocks. Thus the expression for the event rendering loss is:

$$\begin{gathered} \mathcal{L}_{event} = \textstyle\sum_{\boldsymbol{x}\in\chi}\left\|\widehat{B}_{(k_1,k_2)}(\boldsymbol{x}) - B_{(k_1,k_2)}(\boldsymbol{x})\right\|_2^2, \\ B_{(k_1,k_2)}(\boldsymbol{x}) = \textstyle\sum_{k=k_1+1}^{k_2} B_k(\boldsymbol{x}), \end{gathered} \tag{12}$$

where χ is the set of pixels in each batch. The final loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{blur} + \omega\mathcal{L}_{event}, \tag{13}$$

where ω is the weight parameter.

V. EXPERIMENT

A. Dataset

Since this experiment improves on the method proposed by E2NeRF, in order to compare with the original model, the dataset constructed by E2NeRF is used in this paper. Based on the original NeRF dataset, camera shakiness was simulated using the camera shakify plugin in Blender. In each view, 17 clear images of the camera under different degrees of shaking were rendered and their corresponding poses were recorded. These 17 images were then input into the simulated event generation model v2e [17] to simulate the event data generated during camera shaking. In addition, in order to obtain the simulated blurred images, the 17 images were processed to the original domain by inverse ISP operation and superimposed, and then ISP processing was utilized to obtain the final blurred images. Each scene contains 100 blurred images and corresponding event data.

The real-world dataset was captured using a DAVIS 346 color event camera. This camera is capable of capturing spatio-temporally aligned event data and RGB frames. Where the resolution of the camera is 346×260 and the exposure events of the RGB frames are set to 100 ms. 5 sets of challenging scenes containing rich color and texture details in low-light environments were acquired with this camera acquisition platform. Each scene contains 30 images with different levels of blurring on different views and the corresponding event data.

B. Experiment Details

The model in this thesis is constructed using 3DGS as pipeline and trained on a single NVIDIA RTX2080Ti GPU for 30,000 iters for each scene. For each hyper-parameter, set $\omega = 1/625$, $b = 4$ the rest of the parameters are kept the same as the default values of 3DGS, and set the positive threshold $\theta_{pos} = 0.2$ and negative threshold $\theta_{neg} = 0.2$ to simulate the generation of events.

For the Blender synthesized data, under each view, 5 poses are selected at equal intervals and sequentially input to the network for rendering. In order to reflect the generalization of the model, experiments were conducted under several synthetic scene data, including challenging scenes such as metallic materials.

For real-world data, since there are only blurred images and corresponding events, one pose estimation model is used to obtain the corresponding five poses. To characterize the high dynamic range of the event data, several scenes were captured under dark light conditions, each containing a total of 30 views.

C. Comparison Study

In order to measure the quality of the Ev3DGS model, we will compare the measurements from the synthetic dataset and under the real-world dataset. In addition this model will be compared with the E2NeRF model which is also based on blurring rendering loss and event rendering loss. The specific experiments are as follows:

Synthetic data. The experiments were performed under 6 synthetic scenes for blur rendering, and PSNR, SSIM and LPIPS were used to evaluate the experimental results, and the Ev3DGS rendering results are shown in Tab. I. The experimental results show that Ev3DGS achieves a good deblurring rendering level in all six scenes, with an average PSNR of 29.96. Especially in the hotdog scene, the PNSR reaches a level of 34.1, which is of very high rendering quality. In addition, in the challenging scenes such as materials and mic, the rendering quality is also good in the face of metallic materials and fine meshes. However, by observing the rendering results of the synthesized scenes in the first line of Fig. 2, we can find that in some scenes, there are some "wrinkles" and "artifacts" on the edges of the objects, which may be due to the sensitivity of the event data to the edges of the moving objects.

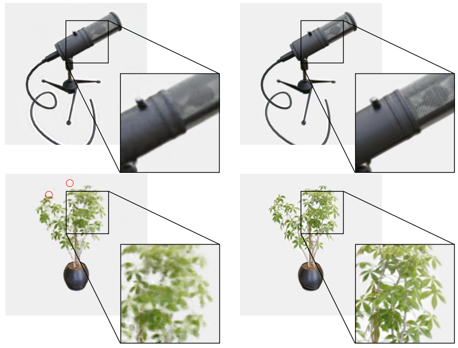

Fig. 2. Comparison of E2NeRF (left) and Our Ev3DGS (right) rendering results on synthetic data of mic (top) and ficus (bottom) scene.

TABLE I
SYNTHETIC DATA RENDERING RESULTS OF EV3DGS METHOD

Ev3DGS	Chair	Ficus	Hotdog	Lego	Materials	Mic	Average
PSNR↑	31.32	30.82	34.10	28.63	27.88	26.96	29.95
SSIM↑	0.9646	0.9654	0.9652	0.9238	0.9378	0.9488	0.9509
LPIPS↓	0.0503	0.0361	0.0610	0.1047	0.0708	0.0545	0.0629

The average training time is about 35min.

TABLE II
SYNTHETIC DATA RENDERING RESULTS OF E2NERF METHOD

E2NeRF	Chair	Ficus	Hotdog	Lego	Materials	Mic	Average
PSNR↑	30.36	25.81	33.43	27.76	27.42	26.61	28.57
SSIM↑	0.9525	0.9035	0.9541	0.8972	0.9255	0.9359	0.9281
LPIPS↓	0.0682	0.0890	0.0562	0.0889	0.0571	0.0831	0.0738

The average training time is about 13h.

The E2NeRF rendering results are shown in Tab. II. The experimental results show that in the same six scenes, the overall rendering level of the two is relatively close. However, the E2NeRF model rendering results show more obvious "patches" and white voids (circle in red), such as those shown in the second line of Fig. 2. In ficus and mic scenes, E2NeRF has a low level of differentiation for dense and small objects, while Ev3DGS has a clearer performance. Under the materials scene, Ev3DGS better reflects the reflection characteristics of metal materials, while E2NeRF's reflection characteristics are less obvious.

Real-world data. Since the real-world data lacks clear images of ground truth, the five real scenes were quantitatively analyzed using the reference-free image quality assessment metrics BRISQUE [18] and RankIQA [19], as shown in Tab. III, and the Ev3DGS results are also better. Under the blur rendering loss and event rendering loss, the event data effectively enhances the perception of 3DGS on dynamic blur. The experimental results show that Ev3DGS also exhibits a better rendering level in low dark real environments thanks to the high dynamic range characteristics of the event camera.

TABLE III
COMPARATIVE RESULTS ON REAL-WORLD DATA

Real-World Data	Ev3DGS	E2NeRF
BRISQUE↓	29.32	29.53
RankIQA↓	3.579	3.588

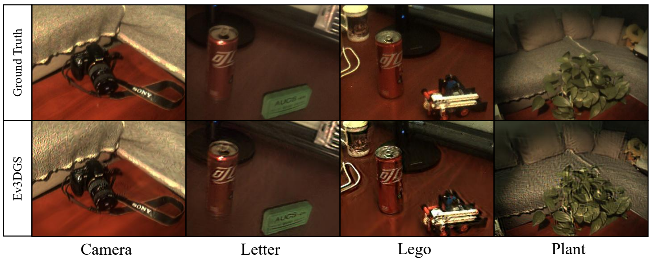

Fig. 3. Real-world data rendering results of Ev3DGS method.

D. Ablation Study

In this experiment, blurry rendering loss and event rendering loss are introduced to the model, and in order to verify the effect of introducing event rendering loss on the model, ablation experiments are carried out based on the synthetic dataset. Where the Ev3DGS model with event loss EventLoss removed is denoted by Ev3DGS*. The experimental results are shown in Tab. IV.

From the results, as shown in Tab. V, the rendered image of Ev3DGS model compared with Ev3DGS* model improves PSNR and SSIM but sacrifices a certain amount of LPIPS.Specifically for the rendering of each scene, it can be found that Event Loss has a large improvement in all the indicators for ficus and hotdog scenes; while for chair, mic scenes For the chair and mic scenes, there is no significant change; for the lego and materials scenes, there is a large improvement in PSNR and SSIM at the expense of certain LIPIS indicators.

TABLE IV
RESULTS OF THE SYNTHETIC DATA OF EV3DGS* METHOD

Ev3DGS*	Chair	Ficus	Hotdog	Lego	Materials	Mic	Average
PSNR↑	31.23	30.70	33.92	28.74	27.57	26.95	29.85
SSIM↑	0.9646	0.9651	0.9648	0.9266	0.9351	0.9487	0.9508
LPIPS↓	0.0496	0.0363	0.0614	0.0998	0.0702	0.0540	0.0618

TABLE V
COMPARATIVE RESULTS OF ABLATION STUDY

Ablation Study	Ev3DGS*	Ev3DGS	Δ
PSNR↑	29.85	29.95	0.34%↑
SSIM↑	0.9508	0.9509	0.01%↑
LPIPS↓	0.0618	0.0629	1.78%↑

VI. CONCLUSIONS

In this paper, a new 3DGS model based on event camera enhancement (Ev3DGS) is proposed, which explores the possibility of combining event camera with 3DGS and provides a solution for 3DGS with blurry input. In addition, the effectiveness of the model is demonstrated experimentally on both synthetic and real datasets. The results show that the Ev3DGS framework has a slight advantage in terms of rendering results over the existing deblurred 3D reconstruction model, the E2NeRF model. However, in terms of training and rendering speed, Ev3DGS is much faster than E2NeRF, which is also due to the ability of fast rendering of 3DGS.In conclusion, the relevant results and conclusions obtained in this paper are useful. It is hoped that the related work in this paper will contribute to the study of high-quality 3D representation learning using event data in complex and low-light scenes.

REFERENCES

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *TOG*, 42(4):1–14, 2023.
[2] Zhexiong Wan, Yuchao Dai, and Yuxin Mao. Learning Dense and Continuous Optical Flow from an Event Camera. *TIP*, 31: 7237–7251, 2022.
[3] Zhexiong Wan, Yuxin Mao, Jing Zhang, and Yuchao Dai. Rpeflow: Multimodal Fusion of RGB-PointCloud-Event for Joint Optical Flow and Scene Flow Estimation. In *ICCV*, pages 10030–10040, 2023.
[4] Juanjuan Zhu, Zhexiong Wan, and Yuchao Dai. Video Frame Prediction from a Single Image and Events. In *AAAI*, volume 38, pages 7748–7756, 2024.
[5] Yunshan Qi, Lin Zhu, Yu Zhang, and Jia Li. E2NeRF: Event Enhanced Neural Radiance Fields from Blurry Images. In *ICCV*, pages 13254–13264, 2023.
[6] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting. In *CVPR*, pages 19595–19604, 2024.
[7] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3D Geometry-aware Deformable Gaussian Splatting for Dynamic View Synthesis. In *CVPR*, pages 8900–8910, 2024.
[8] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. SC-GS: Sparse-Controlled Gaussian Splatting for Editable Dynamic Scenes. In *CVPR*, pages 4220–4230, 2024.
[9] Jeongtaek Oh, Jaeyoung Chung, Dongwoo Lee, and Kyoung Mu Lee. DeblurGS: Gaussian Splatting for Camera Motion Blur. *arXiv preprint arXiv:2404.11358*, 2024.
[10] Lingzhe Zhao, Peng Wang, and Peidong Liu. BAD-Gaussians: Bundle Adjusted Deblur Gaussian Splatting. *arXiv preprint arXiv:2403.11831*, 2024.
[11] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-Based Vision: A Survey. *TPAMI*, 44(1):154–180, 2020.
[12] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras. In *RSS*, 2018.
[13] Chengxi Ye, Anton Mitrokhin, Cornelia Fermüller, James A. Yorke, and Yiannis Aloimonos. Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors. In *IROS*, 2020.
[14] Christian Brandli, Thomas A Mantel, Marco Hutter, Markus A Höpflinger, Raphael Berner, Roland Siegwart, and Tobi Delbruck. Adaptive Pulsed Laser Line Extraction for Terrain Reconstruction Using a Dynamic Vision Sensor. *Frontiers in neuroscience*, 7:275, 2014.
[15] Nathan Matsuda, Oliver Cossairt, and Mohit Gupta. MC3D: Motion Contrast 3D Scanning. In *ICCP*, pages 1–10, 2015.
[16] Liyuan Pan, Richard Hartley, Cedric Scheerlinck, Miaomiao Liu, Xin Yu, and Yuchao Dai. High Frame Rate Video Reconstruction Based on an Event Camera. *TPAMI*, 44(5):2519–2533, 2020.
[17] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From Video Frames to Realistic DVS Events. In *CVPR*, pages 1312–1321, 2021.
[18] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-Reference Image Quality Assessment in the Spatial Domain. *TIP*, 21(12):4695–4708, 2012.
[19] Xialei Liu, Joost Van De Weijer, and Andrew D. Bagdanov. RankIQA: Learning from Rankings for No-Reference Image Quality Assessment. In *ICCV*, pages 1040–1049, 2017.