This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

E-BARF: Bundle Adjusting Neural Radiance Fields from a Moving Event Camera

Zhipeng Tang[1] Shifan Zhu[1] Zezhou Cheng[2] Donghyun Kim[1] Erik Learned-Miller[1]
[1]University of Massachusetts Amherst [2]University of Virginia

Abstract

Neural radiance field (NeRF) training typically requires high-quality images with precise camera poses. These images are often captured with stationary cameras under favorable lighting conditions. Such constraints limit the use of NeRF models in real-world scenarios. Event cameras, a novel neuromorphic vision sensor, can capture visual information with microsecond temporal resolution in challenging lighting conditions, overcoming conventional camera limitations. Recent research has proposed methods to construct NeRFs from fast-moving event cameras, enabling neural radiance field modeling with handheld cameras in high-dynamic-range environments. However, current methods still rely on precise camera poses as input. To obtain this information, these approaches resort to external sensors like motion capture systems, introducing additional complexities such as sensor synchronization, calibration, and increased device costs. While existing camera pose estimation algorithms could potentially replace external sensors, they lack the accuracy needed for high-quality NeRF synthesis, resulting in blurry and low-quality novel views. To address these challenges, we propose E-BARF, a method that simultaneously trains a NeRF model and adjusts the estimated camera poses. This approach achieves high-quality NeRF reconstruction without relying on external sensors, simplifying the NeRF creation process for event cameras.

1. Introduction

Neural Radiance Fields (NeRFs), originally proposed for photorealistic novel view synthesis, have demonstrated their strong ability to implicitly encode 3D scenes using neural network representations [31]. However, training the original NeRF model requires a set of images free from motion blur, along with precise six-degree-of-freedom (6-DoF) camera poses (position and orientation) corresponding to each image's capture location. In practice, to obtain such high-quality training data, stationary cameras are used to capture images in a controlled, well-lit environment, ensuring ideal lighting conditions and avoiding motion blur. These images are then processed using structure-from-motion (SfM) tools such as COLMAP [37–39] for accurate pose estimation. These stringent requirements prevented the deployment of the original NeRF model in applications for general users. Consequently, significant research efforts have been directed toward relaxing requirements related to lighting conditions [14, 19, 32], motion blur [30, 41], and the necessity for accurate camera pose information [18, 21, 27, 43, 44].

One line of research proposes using novel sensors to relax the strict requirements of the original NeRF. Among these sensors, event cameras—a novel type of camera with different working principles from conventional frame-based cameras—have sparked researchers' interest. In 2023, three groups of researchers, Klenk *et al.* [23], Rudnev *et al.* [36], and Hwang *et al.* [16] independently introduced E-NeRF, EventNeRF, and Ev-NeRF, respectively, for training NeRF models using event cameras. They propose novel losses for NeRF training based on event cameras' working principles. Among them, E-NeRF demonstrated its capability to train NeRF models for real-world challenging scenes with high-dynamic-range lighting using a handheld event camera.

The success of E-NeRF, EventNeRF, and Ev-NeRF can be largely attributed to the unique advantages of event cameras. Each pixel operates independently in event cameras by continuously sensing changes in brightness. Once the brightness change exceeds a certain threshold, the pixel emits an event indicating the pixel location, brightness change direction and time of change [6]. Such a design not only allows the event camera to continuously output the sensed visual signals without waiting for a fixed exposure period, but also enables the camera to function effectively in high dynamic range (HDR) environments.[1] As a result, event cameras can handle challenging lighting conditions and mitigate motion blur issues, making them suitable for training NeRF models in complex scenes where conven-

[1]Unlike conventional frame-based cameras where integrating over a fixed time period can lead to saturation, the time period of integration for an event camera can be much shorter or longer, greatly alleviating problems of overexposure and underexposure.

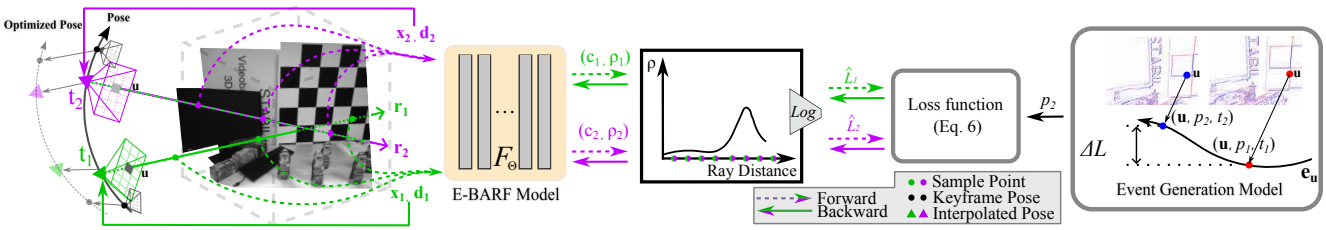


Figure 1. **Overview of E-BARF training.** The event camera's pixel $\mathbf{u}$ triggered two consecutive events at timestamp t_1 and t_2, respectively. The camera poses at timestamp t_1 and t_2 are interpolated from the neighboring keyframe poses (black dots). Rays $\mathbf{r}_1$ and $\mathbf{r}_2$ are the viewing directions of pixel $\mathbf{u}$ at these two timestamps. Each ray goes through the standard NeRF process: For ray $\mathbf{r}_1$, M points (green dots) are sampled along the ray. The brightness value c_1 and volume density ρ_1 of each sampled point are inferred from a NeRF model. The M pairs of (c_1, ρ_1) are then integrated by volume rendering into the pixel value of $\mathbf{u}$ at timestamp t_1, which will be transformed using the linlog function (Eq. 2) to obtain intensity value $\hat{L}_1$. Similarly, we estimate $\hat{L}_2$ with $\mathbf{r}_2$, the pixel value of $\mathbf{u}$ at timestamp t_2. The loss function is formulated based on the event generation model: the brightness change between timestamp t_1 and t_2 at pixel $\mathbf{u}$ should be approximately equal to the contrast threshold C. The backpropagation flows from the loss function back to the keyframe poses.

tional cameras would fail.

Like the original NeRF, these three event-driven NeRFs require accurate event camera poses. However, all of these methods require additional hardware beyond event cameras to obtain these poses: Ev-NeRF uses a conventional frame-based camera and SfM tools. EventNeRF uses a turntable setup in a controlled environment so that the poses can be mathematically inferred, and E-NeRF relies on an expensive motion capture (MoCap) system. The reliance on additional hardware not only increases the overall cost of the solution but also introduces additional operations, such as temporal synchronization and calibration between sensors. These operations may introduce errors that could degrade the quality of the resulting NeRF model.

Our goal is to build NeRFs using solely event camera data. One potential approach is to generate images from events using methods such as E2VID [35], and then use these images as input for training conventional NeRF models. Such ideas were explored by Klenk *et al.* [23] and Rudnev *et al.* [36] in their E-NeRF and EventNeRF works, respectively, and both studies concluded that such an approach was significantly inferior to their own methods, particularly in challenging scenarios. Another possible solution is to estimate the camera poses with algorithms and then utilize the estimated poses for training event-based NeRFs. However, as experiments will show, existing event-based pose estimators are not yet capable of producing trajectories with sufficient accuracy to prevent artifacts in NeRFs. This leads us to the challenge of constructing NeRFs with inaccurate trajectories.

In this paper, we propose Bundle Adjusting Neural Radiance Fields from a moving Event Camera (E-BARF). E-BARF takes as input a stream of events generated by a moving event camera and its imperfect trajectory, and jointly learns the NeRF model from scratch while adjusting the camera trajectory. The output is a NeRF model capable of synthesizing high-quality novel views. More specifically, we first feed the input event stream into a camera pose estimation algorithm to obtain initial, inaccurate estimates of the camera trajectory. The output from these pose estimation methods consists of a series of timestamped poses, referred to as *keyframe poses*, which have significantly lower temporal density compared to the high-frequency event data. For events occurring at intermediate timestamps, corresponding camera poses are computed by interpolating between neighboring keyframe poses. In E-BARF, these keyframe poses are treated as optimizable parameters. By back-propagating the loss from the event-based NeRF model to these pose parameters, E-BARF simultaneously optimizes both the NeRF model and the keyframe poses, thus refining and controlling the trajectory of the event camera. Experiments demonstrate that E-BARF can successfully train a high-quality NeRF model using only a single handheld moving event camera, without relying on any additional sensors for camera pose measurement.

In summary, we present a method that can produce a high-quality NeRF model from nothing but event camera data. In particular, no additional hardware or devices are needed to estimate camera poses, a first in this literature.

2. Related Work

Neural Radiance Fields (NeRFs) [31], developed for photorealistic novel view synthesis, are highly effective at encoding 3D scenes into compact implicit representations, finding applications in various fields like medical imaging [12, 17, 42] and content creation [2, 10, 15, 25]. However, the original NeRF models rely on high-quality training images and accurate camera poses [7]. Some previous works have aimed at addressing challenging lighting conditions [14, 19, 32] and motion blur [30, 41]. Some incorporate new sensors to enhance NeRF's quality and lower its training and deployment requirements [4, 8, 13, 16, 20, 23,

28, 36, 45, 46]. Recently, event cameras gained attention due to their high temporal resolution, dynamic range, and low latency [6], making them ideal for addressing lighting and motion blur issues for NeRFs. In 2023, E-NeRF [23], EvNeRF [16], and EventNeRF [36] were independently proposed to train NeRF using event streams.

However, these event-based NeRF approaches require accurate camera poses for training: EvNeRF obtains poses using SfM algorithms with images [37, 39], E-NeRF relies on a MoCap system, and EventNeRF keeps the camera stationary while rotating objects on a turntable to calculate poses. These solutions either limit the methods' applicability or increase their usage complexity, motivating us to propose E-BARF, a method that builds NeRF with only a moving event camera without reliance on other sensors.

Much research has been conducted to reduce NeRFs' reliance on accurate camera poses [1, 5, 18, 21, 27, 43, 44]. Among them, BARF jointly optimizes NeRF parameters and camera pose registration using a coarse-to-fine strategy [27]. While E-BARF operates in a bundle adjustment manner like BARF, it tackles a more complex problem by optimizing a nearly continuous event camera trajectory rather than a discrete set of camera poses.

Several works following event camera-based NeRFs have been proposed with different goals than E-BARF: Low *et al.* enhanced E-NeRF's robustness by incorporating a more realistic event generation model that accounts for various intrinsic parameters and non-idealities [29]. Qi *et al.* introduced a NeRF-based image deblurring method by extracting motion information from events [33]. Li *et al.* proposed BeNeRF, which trains NeRF using a blurry image and events. [26]. In contrast, E-BARF focuses on eliminating dependencies on additional sensors and aims to push the boundaries of using only an event camera for NeRF training.

3. Approach

3.1. Problem Definition

The goal of E-BARF is to train a NeRF model for a 3D scene[2] only inputting a continuous event stream recorded by an event camera moving around the scene.

The input event stream is a sequence of events. Each event is a triplet $e = (\mathbf{u}, t, p)$ indicating that the brightness sensed by pixel $\mathbf{u}$ increases ($p = +1$) or decreases ($p = -1$) by a threshold at timestamp t. More technically, we use the same event generation model as E-NeRF [23]: Each pixel operates the same asynchronously. For a specific pixel $\mathbf{u}$, assume the k-th event generated by this pixel is $e_k^{\mathbf{u}} = (\mathbf{u}, t_k, p_k)$. This event indicates that, at timestamp t_k, the logarithmic brightness $L(\mathbf{u}, t_k)$ sensed by the pixel $\mathbf{u}$ increases ($p = +1$) or decreases ($p = -1$) by a contrast threshold C, i.e. the event $e_k^{\mathbf{u}}$ is emitted when

$$\Delta L = L(\mathbf{u}, t_k) - L(\mathbf{u}, t_{k-1}) = p_k\, C \tag{1}$$

where $L(\mathbf{u}, t_{k-1})$ is the sensed logarithmic brightness of pixel $\mathbf{u}$ at t_{k-1}, the timestamp at which the last event triggered by the same pixel $\mathbf{u}$ before t_k. During the timespan between t_{k-1} and t_k, there might be brightness change at pixel $\mathbf{u}$, but it does not exceed the threshold $p_k\, C$ since no other events were triggered. Notice that in real event cameras, C changes adaptively with the scene and over time [3].

The logarithmic brightness L is defined as Eq. 2, where $I(\mathbf{u})$ is the light intensity at the pixel $\mathbf{u}$, B is a constant defining the linear region of the function.

$$L(\mathbf{u}, t) = \text{linlog}(I(\mathbf{u})) = \begin{cases} I(\mathbf{u}) \cdot \ln(B)/B, & \text{if } I(\mathbf{u}) < B \\ \ln(I(\mathbf{u})), & \text{else.} \end{cases} \tag{2}$$

The output of E-BARF, a NeRF model, is a neural network trained to estimate the light emitted by a particle in the scene, given its location in the space and the viewing direction in which the particle is observed. To synthesize a novel view, light rays are traced for each pixel of the target view. The pixel values are then computed through volume rendering the particles that lie along the corresponding traced rays, which involves evaluating the particles using the trained NeRF neural network [31, 40]. In E-BARF, such neural network is parameterized by an MLP $F_\Theta : (\mathbf{x}, \mathbf{d}) \to (c, \rho)$, where Θ is the parameters of the MLP, $\mathbf{x} \in \mathbb{R}^3$ is the Cartesian coordinates for the location of the particle, $\mathbf{d} \in \mathbb{S}^2$ is the viewing direction, $c \in \mathbb{R}$ is the predicted brightness of the particle, and ρ is the volume density of the particle (the opaqueness of the particle). Different from the original NeRF, c is not predicted in color. This is because current event camera is majorly monochromatic. Though E-BARF can be extended to color space, there is no mature dataset for training it.

3.2. E-BARF

E-BARF begins by preprocessing the input event stream using an off-the-shelf event camera pose estimation algorithm. Such algorithms typically take three inputs: an event stream, the pre-calibrated intrinsic parameters $\mathbf{K}$, and distortion parameters κ of the event camera. The output is a series of N timestamped event camera poses (referred to as keyframe poses): $\{(t_i, \mathbf{p}_i, \boldsymbol{q}_i)\}_{i=0}^{N-1}$, where t_i is the timestamp for the i-th pose, $\mathbf{p}_i$ is the 3-dimensional translation vector for position, $\boldsymbol{q}_i$ is the quaternion represneting the rotation. Camera poses at intermediate timestamps, say t, between two keyframe poses, say $(t_1, \mathbf{p}_1, \boldsymbol{q}_1)$ and $(t_2, \mathbf{p}_2, \boldsymbol{q}_2)$, can then be obtained through interpolation:

$$\text{Interp}(t) = (t,\ \text{LERP}(\mathbf{p}_1, \mathbf{p}_2, u),\ \text{SLERP}(\boldsymbol{q}_1, \boldsymbol{q}_2, u)) \tag{3}$$

[2] As with NeRF, we assume the scene has no moving objects.

where $u = \frac{t-t_1}{t_2-t_1}$, $t_1 = max\{t_i | t_i \leq t\}$, and $t_2 = min\{t_i | t_i > t\}$.

The forward process for training E-BARF proceeds as follows. From the event stream, we randomly select a pair of events $e_k^{\mathbf{u}} = (\mathbf{u}, t_k, p_k)$ and $e_{k-1}^{\mathbf{u}} = (\mathbf{u}, t_{k-1}, p_{k-1})$[3], which were triggered from the same pixel $\mathbf{u}$. The second event $e_{k-1}^{\mathbf{u}}$ is the latest event at pixel $\mathbf{u}$ that had been triggered before t_k.

We begin by rendering the intensity value of pixel $\mathbf{u}$ at timestamp t_k: The camera pose at timestamp t_k is interpolated as $\mathbf{T}_{e_k^{\mathbf{u}}} = \text{Interp}(t_k)$. The ray $\mathbf{r}_k$ that forms the event is then computed via back-projection as $\mathbf{r}_k = \pi^{-1}(\mathbf{T}_{e_k^{\mathbf{u}}}, \mathbf{u}, \mathbf{K}, \kappa)$.

NeRF synthesizes the pixel value at $\mathbf{u}$ via volume rendering: for each particle $\mathbf{x}$ that lies along the ray $\mathbf{r}_k$, NeRF estimates its brightness c and volume density ρ through $F_\Theta(\mathbf{x}, \mathbf{d}) = (c, \rho)$ where $\mathbf{d}$ is the viewing direction.[4] Volume rendering formula integral over all such particles to get the final pixel value. In practice, such integral formulation is approximated via quadrature on discrete number of particles [27]. E-BARF follows the practice in E-NeRF, and perform uniform sampling along the ray $\mathbf{r}_k$: $\{\mathbf{x}_i^{\mathbf{r}_k} | _{i=0}^{M-1}\}$. For simplicity, we do not delve into the details of volume rendering (see [40] for reference) but just summarize the approximated volume rendering formula into a function, which takes the NeRF-estimated c and ρ of M particles sampled along the ray $\mathbf{r}_k$, and outputs the estimated value of pixel $\mathbf{u}$ at timestamp t_k.

$$\hat{I}_{t_k}(\mathbf{u}) = \text{VolumeRender}\left(\left\{F_\Theta\left(\mathbf{x}_i^{\mathbf{r}_k}, \mathbf{d}\right)\right\}\Big|_{i=0}^{M-1}\right) \quad (4)$$

Similarly, we can compute $\hat{I}_{t_{k-1}}(\mathbf{u})$, the intensity value of pixel $\mathbf{u}$ at timestamp t_{k-1}. Based on the event generation model (sec. 3.1), we know that event $e_k^{\mathbf{u}}$ was triggered because the logarithmic brightness change exceeds C since t_{k-1}, the timestamp of the last event triggered in pixel $\mathbf{u}$ before timestamp t_k. Based on NeRF, we know the brightness change between t_k and t_{k-1} is $\Delta\hat{L}_k = \text{linlog}(\hat{I}_{t_k}) - \text{linlog}(\hat{I}_{t_{k-1}})$. Thus, we get a loss function:

$$\mathcal{L}\left(\mathbf{p}_i|_{i=0}^{N-1}, \boldsymbol{q}_i|_{i=0}^{N-1}, \Theta\right) = \left\|\Delta\hat{L}_k - p_k C\right\|_2^2 \quad (5)$$

As the contrast threshold C varies over time, followed E-NeRF, we use the following loss function instead:

$$\mathcal{L}_{\text{norm}}\left(\mathbf{p}_i|_{i=0}^{N-1}, \boldsymbol{q}_i|_{i=0}^{N-1}, \Theta\right) = \left\|\frac{\Delta\hat{L}_k}{\|\Delta\hat{L}_k\|_2^2} - p_k\right\|_2^2 \quad (6)$$

[3]For simplicity, we only describe one event pair, but the forward process is batchified with a number of randomly selected input event pairs.

[4]We encode $(\mathbf{x}, \mathbf{d})$ with multi-resolution hash encoding. We omit this detail here for simplicity.

Unlike E-NeRF, E-BARF treats the event camera poses as optimizable parameters, refining their position and orientation through back-propagation. This approach enables joint learning of the NeRF model from scratch while simultaneously refining the estimated trajectory to maximize the number of event pairs that align with the event generation model.

4. Experiments

We validate the effectiveness of E-BARF with real-world datasets (TUMVIE [22], EDS [11], and EVO [34] dataset) because synthetic data generated by event camera simulators have limitations in accurately modeling certain properties of real event cameras, such as the adaptive contrast threshold C (sec. 3.1). An algorithm that performs well in simulation does not necessarily work as effectively in real-world environments. These three datasets include diverse lighting conditions and use different physical event cameras, demonstrating E-BARF's broad applicability. We perform direct comparisons between E-BARF and E-NeRF. However, comparisons with EventNeRF are not feasible due to its limited support for operating in real-world scenarios and its assumption of a constant color background in the scene. Additionally, we cannot evaluate against EvNeRF as their implementation has not been made publicly available.

4.1. Datasets

In this section, we introduce the three datasets used in our experiments: TUMVIE, EDS, and EVO dataset. These datasets are widely used for evaluating event camera pose estimation algorithms, allowing us to readily adopt existing algorithms for trajectory estimation, the first step of E-BARF. Both TUMVIE and EDS were previously used in E-NeRF experiments, enabling direct comparisons.

The TUMVIE dataset is captured with a sensor rig equipped with high-resolution event cameras, frame-based cameras, and an IMU [22]. IR-reflective passive markers are attached to the sensor rig, enabling the MoCap system to track its pose at a frequency of 120 Hz. The timestamps of all sensors are synchronized in hardware. As a result, each sequence includes a synchronized sequence of events, images, IMU data, and sensor rig poses. The authors carefully calibrated all sensors, which allows for the derivation of both event camera and conventional frame-based camera trajectories (expressed as 120 Hz camera poses) from the rig poses. Inevitably, potential errors in the MoCap system, such as spatial discrepancies between the marker frame and optical center, calibration inaccuracies in event camera intrinsic parameters, and temporal misalignment between the MoCap and event camera, can cause inferred event camera poses to deviate slightly from the true trajectory. Therefore, we refer to these trajectories as *pseudo-ground truths*.

Similar to TUMVIE, the EDS dataset also provides synchronized event stream, images, IMU, and ground truth poses from a MoCap system (at 150 Hz) [11]. Unlike TUMVIE, EDS dataset is recorded with a beamsplitter to guarantee the conventional frame-based camera and the event camera share the same optical axis.

The EVO dataset [34] was released together with the pioneering event-based visual odometry with the same name. The sequences in this dataset include events and images captured by an early DAVIS event camera with 240×180 resolution. Unlike TUMVIE and EDS, the EVO dataset is not recorded with a MoCap system and therefore lacks pseudo-ground truth trajectories.

4.2. Analysis on Synthetic Noise

E-BARF is proposed to train an accurate NeRF model with events given inaccurate event camera poses. We aim to quantitatively and qualitatively analyze how pose inaccuracy affects the performance of E-NeRF and how E-BARF addresses this challenge to achieve high-quality image synthesis. To facilitate such systematic analysis, we controllably perturb the pseudo-ground truth with Gaussian noise to simulate the inaccurate camera poses estimated by real-world algorithms. These tampered trajectories serve as the starting point for training both E-NeRF and E-BARF. Specifically, the pseudo-ground truth trajectory consists of a sequence of timestamped keyframe poses at a fixed frequency (e.g. 120 Hz in TUMVIE). Since event cameras operate at much higher temporal resolutions than the keyframe pose frequency, the pose of the event camera at an arbitrary timestamp is interpolated from neighboring keyframe camera poses (Eq. 3). Consequently, perturbations introduced to keyframe poses directly propagate to temporally adjacent event poses. We introduce this perturbation by applying Gaussian noise to each keyframe camera pose using a Lie algebra drawn from a Gaussian distribution with standard deviation σ, i.e. $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{I_6})$ where $\mathbf{n} \in \mathbb{R}^6$ denotes the noise vector in the Lie algebra space of $\mathbf{SE}(3)$.

We adopt the sequences from TUMVIE. Unlike EDS, which features drastic camera motion and highly challenging lighting, TUMVIE provides clearer ground-truth images, enabling quantitative visual comparison. Meanwhile, the EVO dataset lacks pseudo-ground truth trajectories, limiting its applicability for trajectory evaluation.

4.2.1. Visual Quality Evaluation

For visual quality evaluation, we qualitatively compare the images captured by the frame-based camera with the views synthesized from the corresponding camera poses of these images. We use widely adopted metrics PSNR, LPIPS, and SSIM to quantify the comparison.

For scene *MoCapDesk2* from TUMVIE, Fig. 2f shows the perturbed trajectory (blue), which serves as the input for the training of E-NeRF and E-BARF. This perturbed trajectory is composed of multiple segment trajectories. The pseudo-ground truth is a continuous trajectory, but we intentionally remove certain segments for testing purposes: events corresponding to these removed segments are excluded from training. We only compare the synthesized views with images captured during the removed segments to prevent potential data leakage that might favor our results.

Fig. 2a and 2b show the synthesized views from an example camera pose using the trained E-NeRF and E-BARF, respectively. For reference, Fig. 2c shows the corresponding image captured by a conventional frame-based camera. It is clear that E-NeRF struggles to synthesize a recognizable view with the perturbed trajectory, whereas E-BARF produces synthesized view similar to Fig. 2c. This suggests that while E-NeRF is significantly affected by the added noise, E-BARF can still train NeRF to generate high-quality images despite the pose errors. Fig. 2d and 2e present the depth maps, demonstrating the trained NeRFs' understanding of scene geometry. The quality of these depth maps aligns with that of the rendered views.

The quantitative metrics align with these qualitative observations (see Table 1): images synthesized from E-NeRF show worse metrics compared to those generated by E-BARF when evaluated against the ground truth images.

Table 1. Metrics for Visual Comparison: Synthesized images from holdout test event camera poses, generated using E-NeRF and E-BARF, are compared to corresponding frame-based camera images, with average PSNR, SSIM, and LPIPS values reported.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓
MoCapDesk2	E-NeRF	9.21	0.28	0.78
	E-BARF	**12.53**	**0.47**	**0.46**
MoCap1dtrans	E-NeRF	11.31	0.26	0.84
	E-BARF	**13.16**	**0.41**	**0.57**

It's worth mentioning that the synthesized view may be slightly misaligned with the ground truth image. This is because, in E-BARF, both the scene and camera poses are variable up to a 3D similarity transformation [27]. As a result, it is difficult to synthesize views from exactly the same camera poses as the test image. However, the commonly adopted visual quality metrics PSNR, SSIM, and LPIPS are sensitive to image misalignment. Although aligning the pseudo-ground truth with the adjusted trajectory could help, perfect alignment is unachievable, so misalignment persists. An alternative approach is to apply image registration before metric evaluation; however, this approach fails when synthesized images seriously degrade due to noisy poses (see Fig. 2a and 2g). To ensure fairness in our evaluation, we opt to leave the images unaltered during comparison.

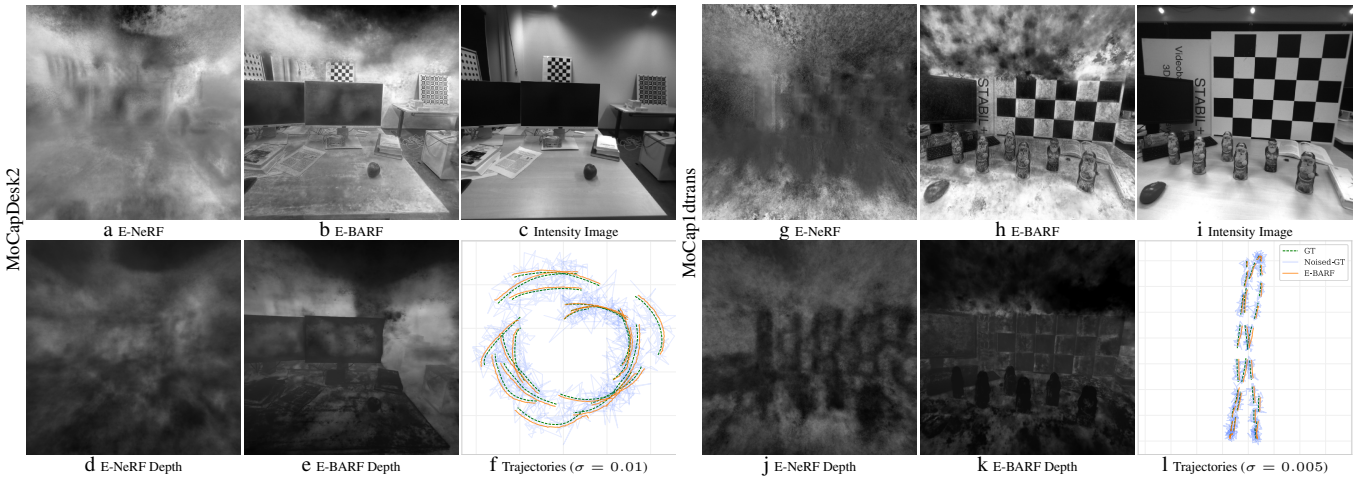


Figure 2. Visual comparison between E-NeRF and E-BARF on scenes *MoCapDesk2* (left) and *MoCap1dtrans* (right): The pseudo-ground truth trajectory (dashed green line in Fig. 2f) was perturbed with Gaussian noise $\mathcal{N}(0, \sigma^2\mathbf{I_6})$, resulting in the perturbed trajectory (blue line in Fig. 2f) used for training both E-NeRF and E-BARF. Novel synthesized views generated by E-NeRF and E-BARF are shown in Fig. 2a and Fig. 2b, respectively, with corresponding depth images presented in Fig. 2d and Fig. 2e. The ground truth image captured by a frame-based camera (Fig. 2c) more closely resembles the view synthesized by E-BARF (Fig. 2b) than that by E-NeRF (Fig. 2a). Similarly, the trajectory refined by E-BARF (orange line in Fig. 2f) aligns more closely with the pseudo-ground truth (dashed green line) compared to the perturbed trajectory (blue line). Results for *MoCap1dtrans*, following the same layout, are presented on the right side (Fig. 2g–Fig. 2l).

Although this decision may lead to metrics that are less favorable for E-BARF (e.g. the absolute metric values are lower than usual), the results still clearly show that E-BARF outperforms E-NeRF.

Fig. 2 (right side) and Table 1 also present qualitative and quantitative experimental results, respectively, for scene *MoCap1dtrans* from TUMVIE. Both results lead to the same conclusion as *MoCapDesk2*.

4.2.2. Trajectory Adjustment Evaluation

Fig. 2f and Fig. 2l qualitatively illustrate the pseudo-ground truth trajectories (dashed green), the perturbed trajectories (blue), and the trajectories adjusted by E-BARF (orange) for scenes *MoCapDesk2* and *MoCap1dtrans*, respectively, demonstrating how E-BARF corrects perturbed trajectories toward the pseudo-ground truth. Despite substantial perturbations, E-BARF refines the perturbed trajectories to align with the shape of pseudo-ground truths.

The first and second rows of Table 2 (marked with †) present the distances, measured using absolute pose error (APE), between the pseudo-ground truth trajectories and both the perturbed trajectories and the E-BARF-adjusted trajectories. The distances between the E-BARF-adjusted trajectories and the pseudo-ground truth trajectories show a notable reduction compared to those between the perturbed trajectories and the pseudo-ground truth trajectories. These quantitative results provide evidence that E-BARF effectively refines perturbed trajectories toward the pseudo-ground truth, supporting the qualitative observations discussed earlier.

As discussed in sec. 4.2.1, the trajectories adjusted by E-BARF only align with the pseudo-ground truth trajectories up to a 3D similarity transformation. Although the controlled perturbation of the pseudo-ground truth trajectories is intended to reduce such transformations, it cannot completely eliminate them (see the misalignments between the refined trajectories and the pseudo-ground truths in Fig. 2f and 2l). Performing trajectory alignment before evaluation may reduce misalignment. However, the perturbed trajectories are too noisy for effective alignment. Additionally, this process inevitably conflates trajectory adjustment errors with alignment errors. Hence, we keep the trajectories unaltered during visualization and metric computation. While this decision may result in distance metrics that are less favorable for E-BARF due to the inclusion of misalignment-induced errors, the results clearly demonstrate that E-BARF adjusts the trajectories to better align with the pseudo-ground truth.

4.3. Experiments on Real Estimated Trajectories

In this section, we evaluate E-BARF's performance when training from trajectories estimated by existing event camera pose estimation algorithms, which represents the intended use case for E-BARF. Specifically, we employ the most recent Deep Event Visual Odometry (DEVO) algorithm [24]. Estimating 6-DOF poses solely from event data is challenging, resulting in limited available algorithms [24]. We also tested the EVO algorithm [34], but it diverged early on in nearly all datasets, leading us to abandon its use. For data sequences that DEVO failed or does

Table 2. Metrics for Trajectory Comparison: The Trajectory column shows the initial trajectories used for training E-NeRF and E-BARF. Noised GT refers to pseudo-ground truth trajectories perturbed by Gaussian noise (blue lines in Fig. 2f and 2l, depending on the scene). DEVO denotes trajectories estimated by the DEVO algorithm. Metrics report the mean of Absolute Pose Error (T_{ape}) between the pseudo-ground truth and post-training trajectories. For E-BARF, these are refined trajectories, while for E-NeRF, they are initial trajectories since it does not modify them during training. Scene *Flyingroom* is not listed due to the lack of pseudo-ground truths for metric computation. † presents experimental results for sec. 4.2.2, and ‡ for sec. 4.3.

Scene	Trajectory	Method	T_{ape} (mean)
MoCapDesk2†	Noised GT	E-NeRF	2.89
		E-BARF	**0.62**
MoCap1dtrans†	Noised GT	E-NeRF	1.45
		E-BARF	**0.60**
MoCapDesk2‡	DEVO	E-NeRF	1.04
		E-BARF	**0.52**
MoCap1dtrans‡	DEVO	E-NeRF	0.78
		E-BARF	**0.62**
EDS11‡	DEVO	E-NeRF	3.51
		E-BARF	**3.31**

not natively support, we explored an alternative approach: converting events to greyscale frames using E2VID [35] and then estimating trajectories using COLMAP.

We conduct qualitative and quantitative analysis on trajectory adjustment. The fourth column of Fig. 3 visualizes the input trajectories estimated by real-world algorithms (blue lines), the E-BARF-refined trajectories (orange lines), and the pseudo-ground truths (dashed green lines). Quantitative evaluation results are presented in Table 2 (marked with ‡). Unlike the experiments in sec. 4.2.2, where input trajectories were perturbed in a controlled manner from pseudo-ground truths, significantly reducing the 3D similarity transformation, the input trajectories estimated by real-world algorithms differ significantly from pseudo-ground truths in scale, rotation, and translation, not to mention estimation errors. These differences propagate to the E-BARF-refined trajectories. To enable visualization and ensure a reasonable quantitative evaluation, trajectory alignment is required. We align and scale the trajectories for comparisons with the pseudo-ground truths using the open-source software *evo* [9]. The APE metric used in quantitative evaluation considers both orientation and position, with the unit-less mean pose errors reported.

Regarding image quality, we focus on qualitative comparisons, as illustrated in the first three columns of Fig. 3. As noted by Klenk *et al.* [23], the experimental scenes present significant challenges, with ground truth grayscale images exhibiting artifacts such as severe motion blur or loss of detail in high dynamic range (HDR) regions. Furthermore, the presence of a non-negligible 3D similarity transformation creates noticeable misalignment between the synthesized views and ground truth images. These factors make the computation of the metrics for visual quantitative comparison both challenging and uninformative.

The first and second rows of Fig. 3 show qualitative results for *MoCapDesk2* and *MoCap1dtrans* from the TUMVIE dataset, respectively. In both cases, synthesized views from E-BARF are sharper and more detailed compared to those from E-NeRF, with significant improvement in *MoCapDesk2* and minor in *MoCap1dtrans*. This difference can be explained by the accuracy of the input trajectories estimated by DEVO. In *MoCapDesk2*, the DEVO-estimated trajectory contains severe errors, making E-NeRF training difficult and resulting in blurry, melted-like synthesized views. E-BARF corrects these errors, refining the trajectory to closely match the pseudo-ground truth and producing higher-quality views. For *MoCap1dtrans*, DEVO provides a more accurate trajectory, enabling E-NeRF to synthesize decent views, while E-BARF makes slight adjustments to enhance sharpness further. The results in Table 2 support the visual analysis, with E-BARF in *MoCapDesk2* showing a more significant advantage over E-NeRF in terms of APE compared to *MoCap1dtrans*. It is worth noting the noticeable misalignment between the E-BARF-synthesized views and the ground truth images in both scenes shown in Fig. 3.

The third row of Fig. 3 presents results for the *EDS11* scene from the EDS dataset. The ground truth image suffers from severe motion blur caused by rapid camera shaking, rendering the view nearly unrecognizable. While E-NeRF partially reduces the blur, E-BARF synthesizes significantly sharper views, recovering most of the details. Quantitative analysis further confirms that the trajectory refined by E-BARF aligns more closely with the pseudo-ground truth than the trajectory estimated by DEVO (see Table 2).

The fourth row of Fig. 3 presents results for the *Flyingroom* scene from the EVO dataset, filmed in an HDR environment where conventional frame-based cameras struggle to capture textures in dark regions. While E-NeRF recovers some lost details, they remain slightly blurry (e.g., the book title). In contrast, E-BARF produces sharper images with more preserved details. The trajectories estimated by E2VID+COLMAP exhibit an unnatural, unsmooth pattern, which is smoothed out by E-BARF. This pattern is likely caused by estimation errors, as the camera motion appears smooth when observing the video captured by the frame-based camera. However, confirming this hypothesis or conducting quantitative analysis is challenging due to the absence of pseudo-ground truths.

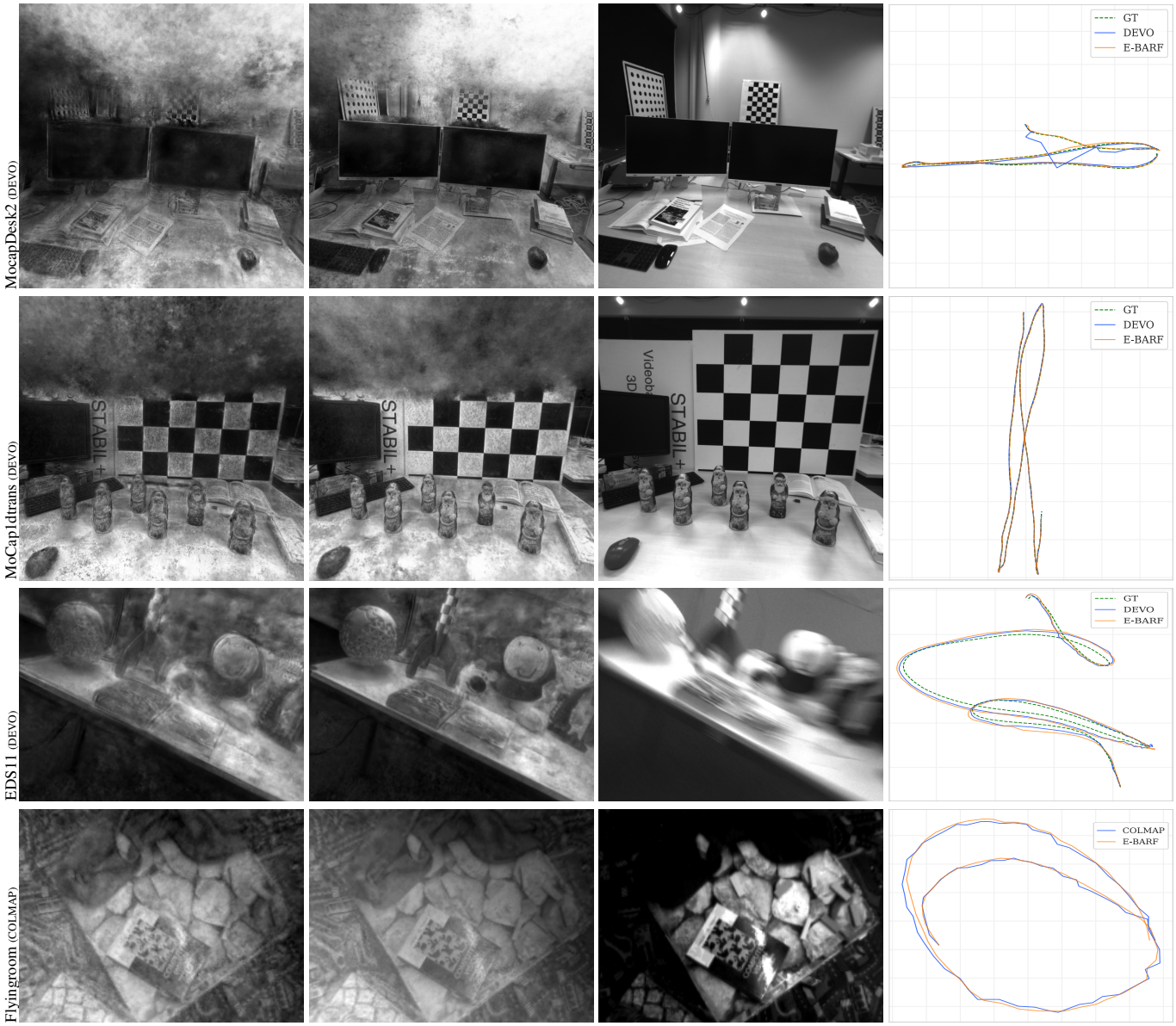


Figure 3. Qualitative comparisons between E-NeRF and E-BARF trained on trajectories estimated by off-the-shelf algorithms: The first and second columns display the synthesized images generated by E-NeRF and E-BARF, respectively. The third column shows the ground truth images captured by frame-based cameras from the corresponding camera poses. The fourth column visualizes the trajectories: dashed green lines represent the pseudo-ground truths, blue lines indicate the initial trajectories used for training, and orange lines depict the trajectories refined by E-BARF. Trajectories are scaled and aligned for comparisons. Each row corresponds to an example camera view from a specific scene (from top to bottom): *MoCapDesk2* and *MoCap1dtrans* from TUMVIE, *EDS11* from EDS, and *Flyingroom* from EVO. The initial trajectories for all scenes were estimated using DEVO, except for *Flyingroom*, which was estimated using E2VID+COLMAP.

5. Conclusions

In this paper, we propose E-BARF, a method that jointly trains NeRF models from scratch while adjusting camera poses, enabling the training of NeRFs with a single hand-held, fast-moving event camera in challenging lighting conditions. Our results show that E-BARF outperforms baseline algorithms across multiple datasets. In the future, we aim to expand it into an event-based structure-from-motion approach.

References

[1] Zezhou Cheng, Carlos Esteves, Varun Jampani, Abhishek Kar, Subhransu Maji, and Ameesh Makadia. Lu-nerf: Scene

and pose estimation by synchronizing local unposed nerfs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18312–18321, 2023. 3

[2] Peng Dai, Feitong Tan, Xin Yu, Yinda Zhang, and Xiaojuan Qi. Go-nerf: Generating virtual objects in neural radiance fields. *arXiv preprint arXiv:2401.05750*, 2024. 2

[3] Tobi Delbruck, Yuhuang Hu, and Zhe He. V2e: From video frames to realistic dvs event camera streams. *arXiv e-prints*, pages arXiv–2006, 2020. 3

[4] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[5] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2, 2024. 3

[6] 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. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1, 3

[7] Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Nerf: Neural radiance field in 3d vision, a comprehensive review. *arXiv preprint arXiv:2210.00379*, 2022. 2

[8] Beerend GA Gerats, Jelmer M Wolterink, and Ivo AMJ Broeders. Nerf-or: neural radiance fields for operating room scene reconstruction from sparse-view rgb-d videos. *International journal of computer assisted radiology and surgery*, pages 1–10, 2024. 2

[9] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017. 7

[10] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 2

[11] Javier Hidalgo-Carrió, Guillermo Gallego, and Davide Scaramuzza. Event-aided direct sparse odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5781–5790, 2022. 4, 5

[12] Jing Hu, Qinrui Fan, Shu Hu, Siwei Lyu, Xi Wu, and Xin Wang. Umednerf: Uncertainty-aware single view volumetric rendering for medical neural radiance fields. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–4. IEEE, 2024. 2

[13] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural lidar fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18236–18246, 2023. 2

[14] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18398–18408, 2022. 1, 2

[15] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18342–18352, 2022. 2

[16] Inwoo Hwang, Junho Kim, and Young Min Kim. Ev-nerf: Event based neural radiance field. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 837–847, 2023. 1, 2, 3

[17] Khadija Iddrisu, Sylwia Malec, and Alessandro Crimi. 3d reconstructions of brain from mri scans using neural radiance fields. In *International Conference on Artificial Intelligence and Soft Computing*, pages 207–218. Springer, 2023. 2

[18] Nishant Jain, Suryansh Kumar, and Luc Van Gool. Learning robust multi-scale representation for neural radiance fields from unposed images. *International Journal of Computer Vision*, 132(4):1310–1335, 2024. 1, 3

[19] Kim Jun-Seong, Kim Yu-Ji, Moon Ye-Bin, and Tae-Hyun Oh. Hdr-plenoxels: Self-calibrating high dynamic range radiance fields. In *European Conference on Computer Vision*, pages 384–401. Springer, 2022. 1, 2

[20] Saimouli Katragadda, Woosik Lee, Yuxiang Peng, Patrick Geneva, Chuchu Chen, Chao Guo, Mingyang Li, and Guoquan Huang. Nerf-vins: A real-time neural radiance field map-based visual-inertial navigation system. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10230–10237. IEEE, 2024. 2

[21] Injae Kim, Minhyuk Choi, and Hyunwoo J Kim. Up-nerf: Unconstrained pose prior-free neural radiance field. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[22] Simon Klenk, Jason Chui, Nikolaus Demmel, and Daniel Cremers. Tum-vie: The tum stereo visual-inertial event dataset. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8601–8608. IEEE, 2021. 4

[23] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-nerf: Neural radiance fields from a moving event camera. *IEEE Robotics and Automation Letters*, 8 (3):1587–1594, 2023. 1, 2, 3, 7

[24] Simon Klenk, Marvin Motzet, Lukas Koestler, and Daniel Cremers. Deep event visual odometry. In *2024 International Conference on 3D Vision (3DV)*, pages 739–749. IEEE, 2024. 6

[25] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 2

[26] Wenpu Li, Pian Wan, Peng Wang, Jinghang Li, Yi Zhou, and Peidong Liu. Benerf: neural radiance fields from a single blurry image and event stream. In *European Conference on Computer Vision*, pages 416–434. Springer, 2025. 3

[27] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021. 1, 3, 4, 5

[28] Yvette Y Lin, Xin-Yi Pan, Sara Fridovich-Keil, and Gordon Wetzstein. Thermalnerf: Thermal radiance fields. In *2024 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2024. 3

[29] Weng Fei Low and Gim Hee Lee. Robust e-nerf: Nerf from sparse & noisy events under non-uniform motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18335–18346, 2023. 3

[30] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12870, 2022. 1, 2

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[32] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. *CVPR*, 2022. 1, 2

[33] Yunshan Qi, Lin Zhu, Yifan Zhao, Nan Bao, and Jia Li. Deblurring neural radiance fields with event-driven bundle adjustment. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9262–9270, 2024. 3

[34] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2016. 4, 5, 6

[35] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019. 2, 7

[36] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Eventnerf: Neural radiance fields from a single colour event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4992–5002, 2023. 1, 2, 3

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 3

[38] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[39] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 1, 3

[40] Andrea Tagliasacchi and Ben Mildenhall. Volume rendering digest (for nerf), 2022. 3, 4

[41] Peng Wang, Lingzhe Zhao, Ruijie Ma, and Peidong Liu. Bad-nerf: Bundle adjusted deblur neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4170–4179, 2023. 1, 2

[42] Xin Wang, Shu Hu, Heng Fan, Hongtu Zhu, and Xin Li. Neural radiance fields in medical imaging: Challenges and next steps. *arXiv preprint arXiv:2402.17797*, 2024. 2

[43] Jiahui Zhang, Fangneng Zhan, Yingchen Yu, Kunhao Liu, Rongliang Wu, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Pose-free neural radiance fields via implicit pose regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3534–3543, 2023. 1, 3

[44] Qi Zhang, Chi Huang, Qian Zhang, Nan Li, and Wei Feng. Learning geometry consistent neural radiance fields from sparse and unposed views. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8508–8517, 2024. 1, 3

[45] Haidong Zhu, Yuyin Sun, Chi Liu, Lu Xia, Jiajia Luo, Nan Qiao, Ram Nevatia, and Cheng-Hao Kuo. Multimodal neural radiance field. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9393–9399. IEEE, 2023. 3

[46] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 3