This ICCV 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.

Deformable Neural Radiance Fields using RGB and Event Cameras

Qi Ma[1] Danda Pani Paudel[1,3] Ajad Chhatkuli[1] Luc Van Gool[1,2,3]
[1]Computer Vision Lab, ETH Zurich [2]VISICS, ESAT/PSI, KU Leuven [3]INSAIT, Sofia University

Abstract

Modeling Neural Radiance Fields for fast-moving deformable objects from visual data alone is a challenging problem. A major issue arises due to the high deformation and low acquisition rates. To address this problem, we propose to use event cameras that offer very fast acquisition of visual change in an asynchronous manner. In this work, we develop a novel method to model the deformable neural radiance fields using RGB and event cameras. The proposed method uses the asynchronous stream of events and calibrated sparse RGB frames. In our setup, the camera pose at the individual events –required to integrate them into the radiance fields– remains unknown. Our method jointly optimizes these poses and the radiance field. This happens efficiently by leveraging the collection of events at once and actively sampling the events during learning. Experiments conducted on both realistically rendered graphics and real-world datasets demonstrate a significant benefit of the proposed method over the state-of-the-art and the compared baseline. This shows a promising direction for modeling deformable neural radiance fields in real-world dynamic scenes. We release our code at: https://qimaqi.github.io/DE-NeRF.github.io/

1. Introduction

Neural Radiance Fields (NeRFs) have shown great success in synthesizing photorealistic images by implicitly representing rigid 3D scenes. Modeling non-rigid scenes in such manner is a much more difficult task. Recently, several methods have been proposed to model dynamic neural radiance fields. They aim to model rather slowly deforming radiance fields [5, 15, 16, 6, 30]. The slow deformation assumption is insufficient in scenarios involving fast-moving objects or, equivalently, low frame- rate cameras. In other words, the existing methods cannot capture fast-deforming radiance fields due to the limited frame rate of RGB cameras. To address this problem, we propose to add an event camera that provides information about the radiance change asynchronously.

Event cameras capture radiance changes, also in the presence of fast motions. However, harnessing this benefit comes with its own challenges, mainly due to (i) the unknown absolute radiance at the event location and (ii) the unknown pose of the camera at the time of the event. The former challenge can be addressed by using a hybrid system of RGB and event cameras. We address the latter challenge of pose determination with a novel method.

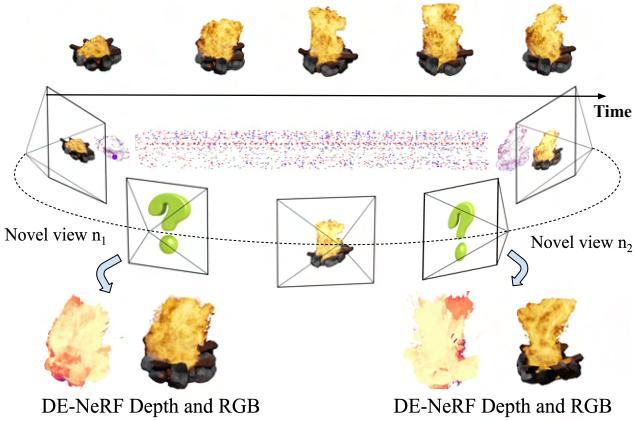

Figure 1. Our framework takes the aligned frames and events captured by a dual RGB-Event camera setup as input. Our method captures fast-moving objects and is capable of rendering a free-viewpoint representation at given timestamps. The figures show the flames' reconstruction with high quality and correct geometry.

Previous methods to deal with the pose of event cameras either do not treat the events to be asynchronous [10, 9] or assume that the event camera's pose is known at all times [21]. We advocate that the event streams must be treated asynchronously to maximally utilize their temporal precision, in keeping with earlier work [23]. On the other hand, we argue that the assumption of known poses for all asynchronous events is simply impractical. Instead, we assume that only the poses of subsequent RGB frames are known. The poses of the events are derived from their associated time stamps, by learning to map time to the evolving camera poses. During this process, the known poses of the RGB frames and the non-rigid deformation prior of the

scene under investigation are jointly utilized.

In this work, we use moving calibrated stereo of RGB and event cameras. Using the known poses of sparse RGB frames only, we want to model the 3D radiance field of deformable objects. To the best of our knowledge, there are thus far no methods leveraging event cameras to model deformable neural radiance fields. Therefore, we first establish a baseline method – which we refer to as DE-baseline – inspired by two notable works on deformable NeRF [15] and event-based NeRF [21]. Later, we propose a novel method that significantly improves this baseline. The proposed method learns to map the time stamp of an event to a camera pose such that each event's ray can be backprojected to the 3D space, without requiring the continuous pose of the asynchronous events. The main idea of this paper is then to constrain the radiance field using the measured events. To do so, we re-create the events solely from the radiance field. Any error due to mismatches between recreated and measured events is backpropagated to supervise the implicit radiance field represention. This radiance field is augmented by sparse and calibrated RGB image frames. The major contributions of this paper are as follows,

- We show the benefit of using event cameras to model the deformable neural radiance fields for the first time.
- We develop a novel method that learns the continuous pose of event cameras which is robust also to inaccurate RGB poses, exploits a collection of events at once, and performs active sampling to maximally utilize the asynchronous event streams.
- The proposed method significantly outperforms existing methods and our baseline on both realistically rendered, but artificial scenes and on real-world datasets.

2. Related Works

Dynamic NeRF: Dynamic NeRFs [17, 15, 16, 25] address the challenging problem of representing static, dynamic or non-rigid scenes using radiance fields [14]. Several works on dynamic NeRF use model-based approaches, e.g., representing human body, hands or faces [30, 31, 33]. Model-free approaches on the other hand, learn a generic deformation function in order to represent non-rigid camera projections or 3D scenes, which is also our interest in this work. Early work D-NERF [17] uses a chosen canonical view to map deformed scenes using a time conditioned function represented by Multi-layer Perceptrons (MLPs). Non-rigidNERF [26] instead deforms the viewing rays and thus the projections instead of the 3D surface, thus, the approach does not directly provide the 3D of the deformed scene. Nerfies [15] train a time conditioned deformation function much like D-NERF [17], albeit with an unknown canonical template-based neural field representation [35]. Furthermore, it also regularizes the deformation field using a coarse-to-fine strategy. As the deformation is defined on 3D space, it can effectively render depths of the non-rigid scene at different time values. HyperNERF [16] introduces shape embeddings in higher dimensions in order to handle topological changes. A very recent work [25] trains NERF for streambale rendering while representing static, rigid and non-rigid scene elements separately.

Event cameras for 3D Vision. Event cameras for 3D reconstruction and camera tracking were presented in [10], based on a probabilistic framework for disparity estimation. Later, [3] addressed camera tracking through a generative modeling of events and maximum likelihood estimation of camera motion. Other contributions have proposed solutions for direct sparse [7] and stereo [37] visual odometry. [18] solves semi-dense multi-view stereo from known poses – by exploiting object silhouettes seen by a moving camera. Similarly, [36] tackles semi-dense stereo-based 3D reconstruction by also solving for the camera motion. [1] reconstructs shapes as a shape from silhouette problem and handles single object reconstruction through synthetic data training. [29] presents a shape from silhouette solution with high quality. Recently, [32] solves non-rigid 3D reconstruction from contours using event cameras. These have also been used for 3D hand pose analysis [22].

Event NeRF. Unlike traditional approaches for 3D reconstruction, NERF-based 3D reconstruction in event cameras is largely under-explored. The generative model-based view synthesis together with surface density estimation in NERF requires highly accurate camera poses and careful optimization, thus rendering its application in event cameras highly challenging. Recent work Event-NERF [21] makes use of a single colour event camera in order to optimize radiance fields, while assuming that the background colour is known in advance. It introduces random temporal window sampling in order to provide diverse supervision. E-NERF [11] presents a NERF method for event frames or events with RGB images in synthetic scenes. The method proposes a normalized loss function in order to handle varying contrast threshold of event cameras. In particular, the method effectively solves deblurring of images using events. Another parallel work Ev-NERF [8] also proposes an event-based NERF method, which uses a threshold-bound loss in order to address the lack of RGB images. The event-to-frame method such as E2VID [20] was employed easily with frame-based NeRF, revealing poor performance that aligns with the findings in our work. All of the previous methods consider the scene to be static, with various assumptions on the contrast threshold of the event camera [8, 21]. A natural question is thus, can event cameras be used to construct NERF to obtain high quality 3D recon-

struction with dynamic objects or scenes, where events can provide a significant edge over conventional cameras? If so, how can we tackle highly challenging non-rigid scenes not addressed by any previous methods? In the following sections we answer these two questions with our proposed method and experiments.

3. Events in the Radiance Field

We represent the pose of the events as a function of time $\mathsf{P}(t)$. At any time t, the 6DoF pose is parameterized by the screw axis $\mathsf{S} = (r(t); v(t)) \in \mathbb{R}^6$ where the rotation matrix and translation vectors can be recovered by Rodrigues's formula [13]. Without loss of generality, we avoid representing the pose of the RGB camera separately. Whenever needed, the RGB camera's pose is related to $\mathsf{P}(t)$ using the known camera extrinsic parameters between the RGB and event cameras. A tuple $\mathbf{e} = (x, t)$ is an event triggered at 2D location x and time t. An event camera measures a set of such tuples, say $\mathcal{E} = \{\mathbf{e}_i\}$. At a sparse set of time stamps, say $\mathcal{T}_s = \{t_j\}$, RGB images with known pose $\mathcal{R} = \{(\mathcal{I}_j, \mathsf{P}(t_j))\}$ are recorded. We are now interested to model the deformable radiance field only using $\mathcal{R}$ and $\mathcal{E}$.

We model the radiance field using the implicit neural representation, with the help of a neural network $\phi_\theta : (\mathsf{X}, \mathsf{d}, t) \rightarrow (\mathsf{c}, \sigma)$ parameterized by θ. Here, any 3D point X, in the world coordinate frame, seen from the viewing direction d at time t is mapped to its color c and density σ. The goal of this paper is to learn θ(from $\mathcal{E}$ and $\mathcal{R}$ with the object deformation prior. We embed the deformation prior in the network architecture. In the following, we first

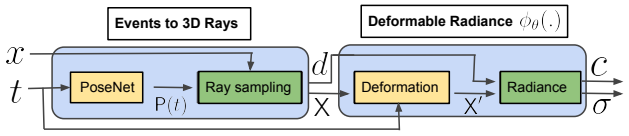

Figure 2. **Events to radiance mapping.** The 2D points x and time t are first mapped to the 3D points X along the viewing direction d, using pose $\mathsf{P}(t)$. Each sampled point is mapped to the canonical space by deformation and decoded into color c and density σ.

3.1. Mapping Events to 3D Rays

Pose from PoseNet. Every event $\mathbf{e} \in \mathcal{E}$ is first mapped to the corresponding pose $\mathsf{P}(t)$, of the camera at the time when the event was triggered. We realize this mapping using the multi-layer perceptron, **PoseNet** as shown in Figure 3 which maps time to screw axis representation $(t) \rightarrow (r; v)$. This neural network generates a continuous pose as a function of time, making it very suitable to handle asynchronous events. The knowledge of the camera pose at the event's

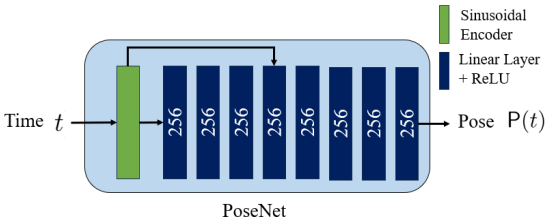

Figure 3. **Time to Pose mapping.** We exploit the implicit neural representation to optimize the camera pose as a continuous function of time. The time $t \in \mathbb{R}$ is firstly normalized to $[-1, 1]$ and then pass to sinusoidal encoder with $L = 10$ of encoded frequencies per axis. The output of network is mapped to rotation and translation using Rodrigues's formula.

time and location allows us to backproject the event into the 3D space represented in the world frame. Unlike other Event-based NeRF that employ trajectory interpolation or turntable poses, we address the joint problem of learning neural 3D representation and refining imperfect event poses similar to[12] .

Sampling event rays. Once the event is backprojected, a set of points are sampled along the ray, as in the standard setting of NeRF training. Then, a trio of a sampled point, viewing direction and event time is formed to infer its radiance and density. Let the 3D point X, direction d, and time t be such a trio. During inferring radiance and density for this trio, the deformation prior is used in network architecture.

3.2. Event Rays in the Deformable Radiance Field

For the deformable radiance field, we assume that there exists a mapping from the deformed surface to a canonical one, as in [35, 15]. Therefore, we first learn to map the 3D point X observed at time t to its canonical position X', by learning the inverse deformation field $\omega(\mathsf{X}, t)$, such that $\mathsf{X}' = \mathsf{X} + \omega(\mathsf{X}, t)$. We realize this inverse deformation field using a multi-layer perceptron. The canonical representation X' is then mapped to the color and density values using another multi-layer perceptron, that additionally receives the viewing direction as input. The arrangement of these two perceptrons, as shown in Figure 2 helps us to realize the deformable radiance mapping network $\phi_\theta : (\mathsf{X}, \mathsf{d}, t) \rightarrow (\mathsf{c}, \sigma)$, where θ is the union of parameters of two sub-networks.

3.3. Rendering Event Ray for Supervision

Let $\mathcal{I}_e \in \mathcal{R}$ be the nearest available RGB image for any event $\mathbf{e} \in \mathcal{E}$. This nearest association is made by comparing the times in tuple (x, t) and the sparse set of time stamps $\mathcal{T}_s$. We then count the effective number of events $n_e = n_e^p - n_e^n$, for n_e^p positive and n_e^n negative number of events which occurred between the time intervals of $\mathcal{I}_e$ and $\mathbf{e}$ acquisitions.

The contrast threshold parameter τ is considered as known. Following the standard volume-rendering [14] strategy, we render the color $\mathcal{I}_{vr}(\mathbf{e})$ for each event. The rendered color $\mathcal{I}_{vr}(\mathbf{e})$ is then compared against the RGB image's color at the event location $\mathcal{I}_e(x)$, while considering the number of intermediate events. More precisely, the event loss for the deformable neural radiance field supervision given by,

$$\mathcal{L}_{event} = \sum_{\mathbf{e}\in\mathcal{E}} \|\mathcal{I}_e(x).\exp(n_e\tau) - \mathcal{I}_{vr}(\mathbf{e})\|, \qquad (1)$$

where τ is the intensity threshold for events to trigger. Note that $\tau = \Delta L/n_e$, for logarithmic change in radiance ΔL. It goes without saying that when the events are only monochromatic, the above loss is computed after accordingly converting the 3-dimensional colors to monochrome.

3.4. Sampling of Events

While using the loss function derived in (1), we employ two strategies for event sampling namely, (i) void and (ii) active. The former aims for better visual consistency whereas the latter improves computational efficiency.

Void sampling. For some arbitrary time stamp t, we randomly select a 2D location x where no event takes place since the last RGB image is acquired. It is intuitive that the color changes minimally for these void events. To impose this constraint, we sample 5% void events and set their effective event count $n_e = 0$. We augment this set of void events to $\mathcal{E}$, while computing the loss of (1).

Active sampling. In the case of the rigid scene and moving camera, it is apparent that the events may not play a significant role in our setup. Instead, they merely introduce the computation burden. The same can be said for the rigid or mostly-rigid parts of the non-rigid scenes. Therefore, we prioritize using events that are generated from the deformable parts. However, such knowledge is not available to us. Therefore, we actively select the desired events during learning. During this, we follow two steps: (i) For $t_j \in \mathcal{T}_s$, we occasionally render the magnitude of the deformation $\omega(\mathsf{X}, t_j)$ from $\mathsf{P}_j \in \mathcal{R}$. (ii) The probability of sampling events near $\mathcal{I}_j$ is then set directly proportional to the rendered deformation magnitude at the event's location.

4. Method Overview

In this section, we present the complete pipeline of our method, as shown in Figure 4. As can be seen, the events are continuously recorded whereas the images are only sparse along the temporal dimension. These sparse images help us to capture the global structure of the radiance field. The finer structures, both in space and time, are then enhanced by using the events. These two aspects however are optimized jointly in an end-to-end manner.

4.1. RGB Cameras for Deformable Fields

We supervise the deformable implicit neural radiance field, $\phi_\theta : (\mathsf{X}, \mathsf{d}, t) \rightarrow (\mathsf{c}, \sigma)$ also using the photometric loss for RGB cameras. When the time-stamp of the calibrated camera is given, the photometric loss is rather straightforward. Let $\mathcal{I}_{vr}(\mathsf{P}_j, t_j)$ be the RGB image rendered from pose P_j for time $t_j \in \mathcal{T}_j$, the photometric loss for RGB cameras is given by,

$$\mathcal{L}_{rgb} = \sum_{\mathcal{I}_j\in\mathcal{R}} \|\mathcal{I}_j - \mathcal{I}_{vr}(\mathsf{P}_j, t_j)\|. \qquad (2)$$

Note that the rendered image is the function of both pose and time-stamp of the corresponding image $\mathcal{I}_j$, as the radiance field is temporally deforming. The above loss function supervises the deforming field only sparsely in time, from the RGB images' poses at the corresponding acquisition time.

4.2. Sparse Poses for Dense Events

Recall that the sparse pairs of camera pose and time (P_j, t_j) for $t_j \in \mathcal{T}_s$ are available in our setting. We use this information in two ways: (i) each P_j is directly used to cast the rays and render the image required for $\mathcal{L}_{rgb}$ computation in (2); (ii) the sparse pairs (P_j, t_j) are used in the time-to-pose mapping network **PoseNet**. Instead of predicting the pose directly from time, we predict the residual pose, where the initial pose for a given event is obtained by local temporal interpolation of available RGB poses.

4.3. The Algorithm

We summarize the loss computation of our method in Algorithm 1. Using the derived loss, three multilayer perceptrons, each for PoseNet, deformation field, and radiance in the canonical frame, are trained. Further implementation details of our method are presented in the next section.

Algorithm 1 $\mathcal{L}_{total}$ = computeTotalLosse($\mathcal{R}, \mathcal{E}, \lambda$)

1. Render the warp field $\omega(\mathsf{X}, t)$ at time t_j from nearby P_j.
2. Sample active events $\mathcal{E}_a \subset \mathcal{E}$ using the rendered wrap field.
3. Sample void events $\mathcal{E}_v$ and set $\mathcal{E}_{total} = \mathcal{E}_a \cup \mathcal{E}_v$.
4. For each event $\mathbf{e} \in \mathcal{E}_{total}$, obtain the pose using **PoseNet**.
5. Render event ray $\mathcal{I}_{vr}(\mathbf{e})$ with event pose and pixel location.
6. Compute event loss $\mathcal{L}_{event}$ for $\mathcal{E}_{total}$ using (1).
7. Sample each ray from $\mathcal{I}_j \in \mathcal{R}$ and render $\mathcal{I}_{vr}(\mathsf{P}_j, t_j)$.
8. Compute the photometric loss $\mathcal{L}_{rgb}$ using (2).
9. Return $\mathcal{L}_{total} = \mathcal{L}_{event} + \lambda\mathcal{L}_{rgb}$

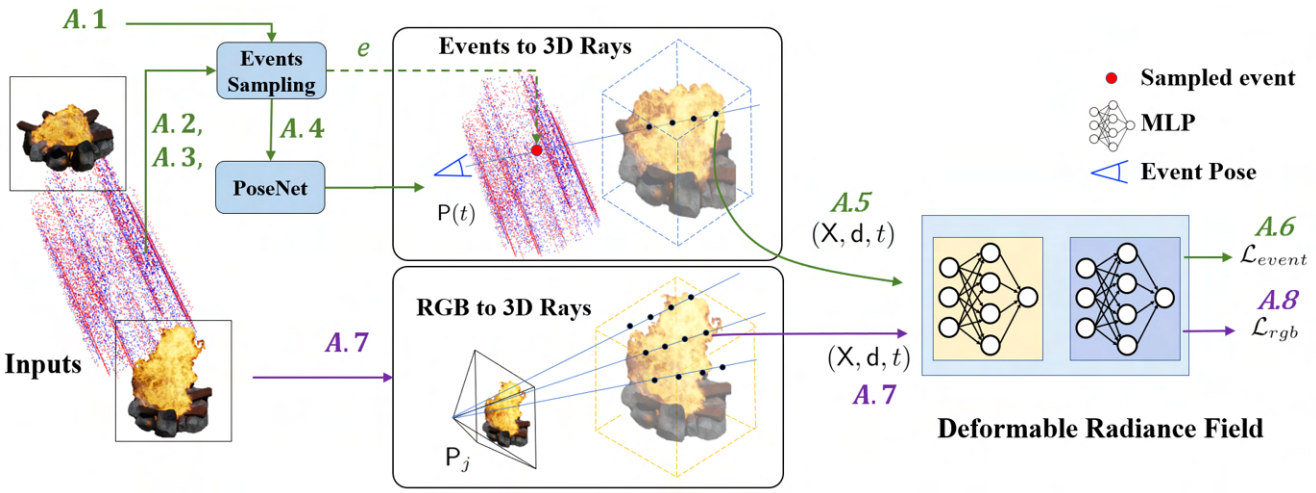

Figure 4. Overview of the proposed method. Notations $A.n$ can be referred to step n accordingly in Algorithm 1.

5. Implementation Details

For the Pose correction network (PoseNet), we use two 8-layer MLPs with the hidden size 256 to learn the translation and rotation residuals. We initialize translation and rotation using cubic spherical linear interpolations, respectively. We use a 6-layer MLPs with a width 128 for the deformation network and output 8 dimension latent codes. We utilize coarse-to-fine regularization to modulate the positional encoding components, as suggested in [15]. We train on 4 NVIDIA GeForce RTX 2080 Ti using 64 coarse rays and 128 fine ray samples. λ=10 is used for RGB loss and in supplementary Tab. 3 we show sensitivity analysis on different λ.

Synthetic Data. Due to the absence of publicly available benchmarks or relevant synthetic event stream datasets and the insufficient number of monocular frames provided by works such as [15] [16][14] to simulate events, we create our own datasets with varying degrees and types of motion using Blender and simulate events using ESIM[19]. We synthesize 3 different scenes. **Non-rigid Lego**: We created a 360-degree camera path around the Lego with the moving 'blade' for two cycles of upward and downward motion which is 4 times faster than [17]. The challenge posed by this dataset is to determine whether the model is capable of effectively acquiring knowledge on the locally-rigid transformation of the blade, and discerning it from the stationary component. **Campfire**: In contrast to Lego, the majority part of the burning campfire dataset is dynamic. The challenge lies in the ability to learn the variations of flame contours and colour from highly varying unordered boundaries. **Fluid**: Water flowing out from pipes hitting the ground with a lot of splashes. The reconstruction of water flow presents the most challenging data, due to its intricate shape variations, colour variations, and the effects of light and shadows. To produce high-temporal resolution events we use Blender to render thousands of continuous frames. In contrast to the approach in [15], which employs a prior based on static 3D background points for regularization, we configure the synthetic dataset's background as white, thereby utilizing it as a means to regularize the background during the training process of the radiance field[14]. All poses are accurate for all synthetic datasets, and the τ of events is also recorded.

Real Data. We evaluate our method on the public datasets which contain dynamic scenes. a) HS-ERGB: We evaluate our method to model deformation on High-Speed Events and RGB dataset [27] which include challenging dynamic scenes such as a rotating **Umbrella** as well as the **Candle** and **Fountain**. Note that in this dataset the camera is static so we disable the PoseNet for residual learning. This dataset provides high-resolution event stream and RGB images. The extrinsics between RGB and event cameras, as well as the pixel-wise alignments, are also provided. b) CED: To evaluate our method on the human subject we use the dynamic **Selfie** sequence in Color Event Camera Dataset [24] which contains both colour frames and colour events from the DAVIS 346C. c) EVIMOv2: We choose the EVIMOv2 Dataset to evaluate our method on the dynamic scenes with moving cameras. The dataset[4] provides millimeter-accurate object poses from a Vicon motion capture system. We use the Samsung DVS Gen3 camera with Flea3 (RGB) as they share most of the field of view. We downsample the rgb frame from 2080×1552 to half and align the frame with events using the depth provided by the Vicon pose estimate and 3D scanning. We selected two se-

quences, namely the **Toycar** with a moderate moving speed and the **Quadcopter** with a high motion speed.

Whenever the contrast threshold τ is not available (or unreliable) for real data, we estimate per-pixel positive and negative thresholds by comparing nearby RGB images and the intermediate event counts [2]. We also filter out mismatched events during this process. For all sequences, we subsample the original high-speed video for training and use the intermediate frames for validation.

Baselines. Since there exists no method that exploits event cameras to model deformable radiance fields, we establish a new baseline method – which we refer to as **DE-baseline**. This baseline is inspired by two notable works on deformable NeRF [15] and event-based NeRF [21]. For DE-Baseline, we sample one ray each for two neighbouring events. Sampled rays are passed through the deformable radiance field, as in our method, using the exact same network. Then, we compute the event loss proposed in [11], together with the photometric loss of (2), for supervision. Similarly to [11] we use the normalized brightness increments loss [7] for real-world cases. In our real data experiments, we found that the normalized event loss is detrimental to PSNR. This aligns with the observation of [11], which can largely be attributed to noisy events.

We also compare our method against two state-of-the-art methods, namely, **Nerfies** [15] and **HyperNeRF** [16], that aim to model the deformable scenes in RGB-only settings. In order to highlight the difficulty, we report the results with the well-known rigid **NeRF** method [14]. Drawing inspiration from other event-based NeRF we also report results using events-to-frame method as reference.

Evaluation Matric. We evaluate our method in learning high-speed dynamic scenes using the following metrics: (i) MSE (with a factor of $\times 10^{-3}$), (ii) Peak signal-to-noise ratio, (iii) The structural similarity (SSIM) [28], and (iv) Learned Perceptual Image Patch Similarity (LPIPS) [34] using VGG. We also follow [38] to calculate the pose error using ATE-RMSE.

6. Experiments

In this section, we provide quantitative and qualitative evaluations. In Table 1 and Table 2, we report the qualitative results for novel view rendering in synthetic and real-world datasets. Our method outperforms all other methods in all MSE, PSNR, SSIM, and LPIPS metrics, thanks to its ability to model fast-moving deformable scenes. This is particularly highlighted in the Lego, Campfire, and Candle datasets. On Campfire, our method not only successfully learns the direction of flame contour changes, with the head of the flame pointing left in the novel view, but also learns

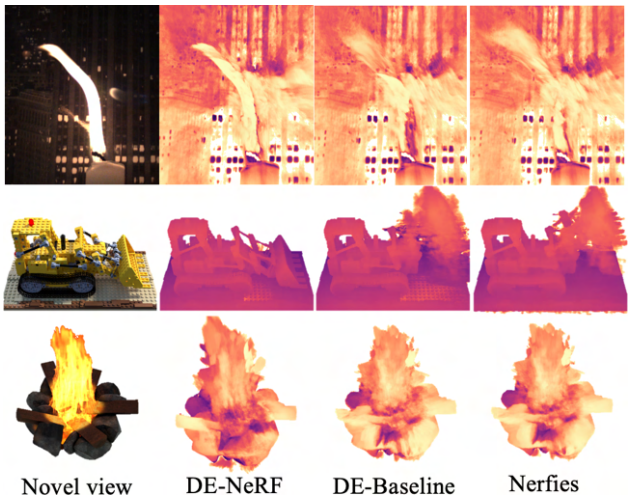

Figure 5. Comparison of rendered depths on three datasets of Tables 1&2. Our method (DE-NeRF) provides very realistic depths.

the depth changes caused by the flame variations. On Candle, our method correctly learns the flame's depth. Similar improvements in performance can be observed in all datasets. Although Fulid is a very challenging (due to the used deformation prior violation), our method still offers a noticeable improvement. Some qualitative results on various datasets are presented in Figure 5 and Table 3. For more visualization please refer to Table 5 in supplementary materials.

The DE-Baseline performs well on synthetic datasets. However, this method needs a large number of samples between the current timestamps and the closest frame. [11] uses a batch size of 30k pairs of events with NVIDIA A40 which is memory and computationally inefficient. In addition, we found that DE-baseline performs numerically worse than Nerfie across all metrics on most real-world data, but with better visual quality. This is similar to the conclusion of the prior work [11].

Additionally, we provide comparison with events-to-frame method [20]. We reconstruct the intensity image and training them together with RGB images using 2. For static camera setup the background trigger no events so we report results only for the segmented dynamic part. As Table 5 shows the events-to-frame based method exhibits poor performance, primarily attributed to challenges such as unknown absolute intensity during reconstruction, as well as domin shifts and artifacts.

It can be observed that our method has limited improvement on fluid and fountain, which is likely due to the difficulty in simulating fluid solely based on the warping radiance field. Furthermore learning depth for water is also challenging due to the shading effect of water. Our method achieved effectively reconstruction of the left person's head

	Lego				Campfire				Fluid			
Methods	MSE ↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓
NeRF [14]	5.54	22.11	0.89	0.229	7.11	21.01	0.85	0.206	3.52	24.94	0.83	0.324
HyperNeRF [16]	4.40	24.28	0.94	0.080	5.27	22.94	0.93	0.152	3.14	25.17	0.85	0.303
Nerfies [15]	2.90	25.97	0.96	0.089	5.13	23.11	0.92	0.154	2.47	25.25	0.87	0.300
DE-Baseline	2.10	27.12	0.97	0.093	4.45	23.76	0.93	0.143	2.51	26.07	0.87	0.296
DE-Nerf(Ours)	**0.32**	**35.04**	**0.99**	**0.034**	**1.95**	**27.56**	**0.96**	**0.115**	**1.91**	**26.92**	**0.91**	**0.289**

Table 1. Comparison of our method against the state-of-the-art and the established baseline, on realistically rendered artificial scenes.

(Static-cameras)	Umbrella				Candle				Fountain			
Methods	MSE ↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓
NeRF [14]	2.72	25.41	0.81	0.471	11.0	19.42	0.86	0.333	6.54	21.89	0.48	0.600
HyperNeRF [16]	2.14	26.72	0.85	0.432	4.01	27.04	0.94	0.283	5.31	22.80	0.52	0.688
Nerfies [15]	1.77	28.30	0.89	0.358	4.23	26.08	0.93	0.293	3.95	24.06	0.66	0.552
DE-Baseline	2.04	27.19	0.86	0.432	4.75	25.72	0.93	0.246	4.13	23.87	0.55	0.610
DE-NeRF (Ours)	**0.45**	**33.44**	**0.95**	**0.341**	**0.38**	**34.22**	**0.97**	**0.242**	**3.11**	**25.13**	**0.71**	**0.546**
(Moving-cameras)	Selfie				Toycar				Quadcopter			
Methods	MSE↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓	MSE↓	PSNR↑	SSIM↑	LPIPS ↓
NeRF [14]	6.25	22.41	0.83	0.382	3.93	24.17	0.85	0.406	7.49	21.25	0.77	0.553
HyperNeRF [16]	3.76	25.02	0.90	0.334	1.41	31.45	0.94	0.242	2.66	27.62	0.92	0.263
Nerfies [15]	2.77	25.85	0.91	0.303	0.87	33.09	0.96	0.217	2.25	28.69	0.93	0.244
DE-Baseline	3.79	24.39	0.89	0.396	1.14	31.99	0.95	0.223	3.01	27.54	0.90	0.265
DE-NeRF (Ours)	**1.80**	**27.74**	**0.94**	**0.224**	**0.54**	**34.17**	**0.98**	**0.201**	**1.53**	**29.95**	**0.95**	**0.210**

Table 2. Real data experiments on two cases: static camera with dynamic scene (top); and moving camera with dynamic scene (bottom). In all six real-world diverse datasets our method performs significantly better than the state-of-the-art methods and established baseline.

movement on the Selfie dataset.

6.1. Ablation Study

We ablate our method for void sampling, active sampling, and pose refinement. The obtained results are presented in Table 4. Our findings indicate that solely relying on with-event location sampling leads to a slight decline in performance. This result may be attributed to the small sampling window utilized in our study (the entire trajectory was divided into 200 windows) , which necessitates that with-event methods provide sufficient information for brightness changes over time. Additionally, our results show that active sampling improves the performance in experiments with 25 and 50 RGB views, with a minimal effect for 10 views. This is expected because for sparse views, the main source of error is triggered by bad events pose estimation from interpolation. As Figure 7 shows adopting active sampling allows for taking advantage of more events triggered by deformation, thereby efficiently learning the warp field. Compared to random sampling, our method achieves more accurate depth in the 10 views case. Finally, our results demonstrate that the use of pose refinement techniques enhances the performance for 10 views cases and leads to further improvements for 25 and 50 views.

6.2. Behaviour Analysis

We conduct several experiments on Lego to investigate the behavior of our method. The performed studies are summarized in Figure 6, with some graphical illustrations in Figure 8. It can be observed that the increasing number of events impacts positively the novel view synthesis as well as the pose estimation, in all cases. At the same time, the lower contrast threshold, or higher sensitivity of the event camera, also leads to better performance, as expected. The pose error in Figure 6 (right) is evaluated using the ATE-RMSE[38]. We also provide the error obtained by initial pose interpolation, for the reference. In Figure 9 we report the rotation error of our method after injecting different magnitudes of rotation noise. We found that our method is robust to small rotation noise and can effectively reduce large rotation noise.

7. Conclusion

In this work, we demonstrated the benefits of event cameras in modelling fast deforming radiance fields. The success of our method is contributed by the proposed novel neural architecture design, training strategy, and the instantaneous nature of the asynchronous event streams. Our ex-

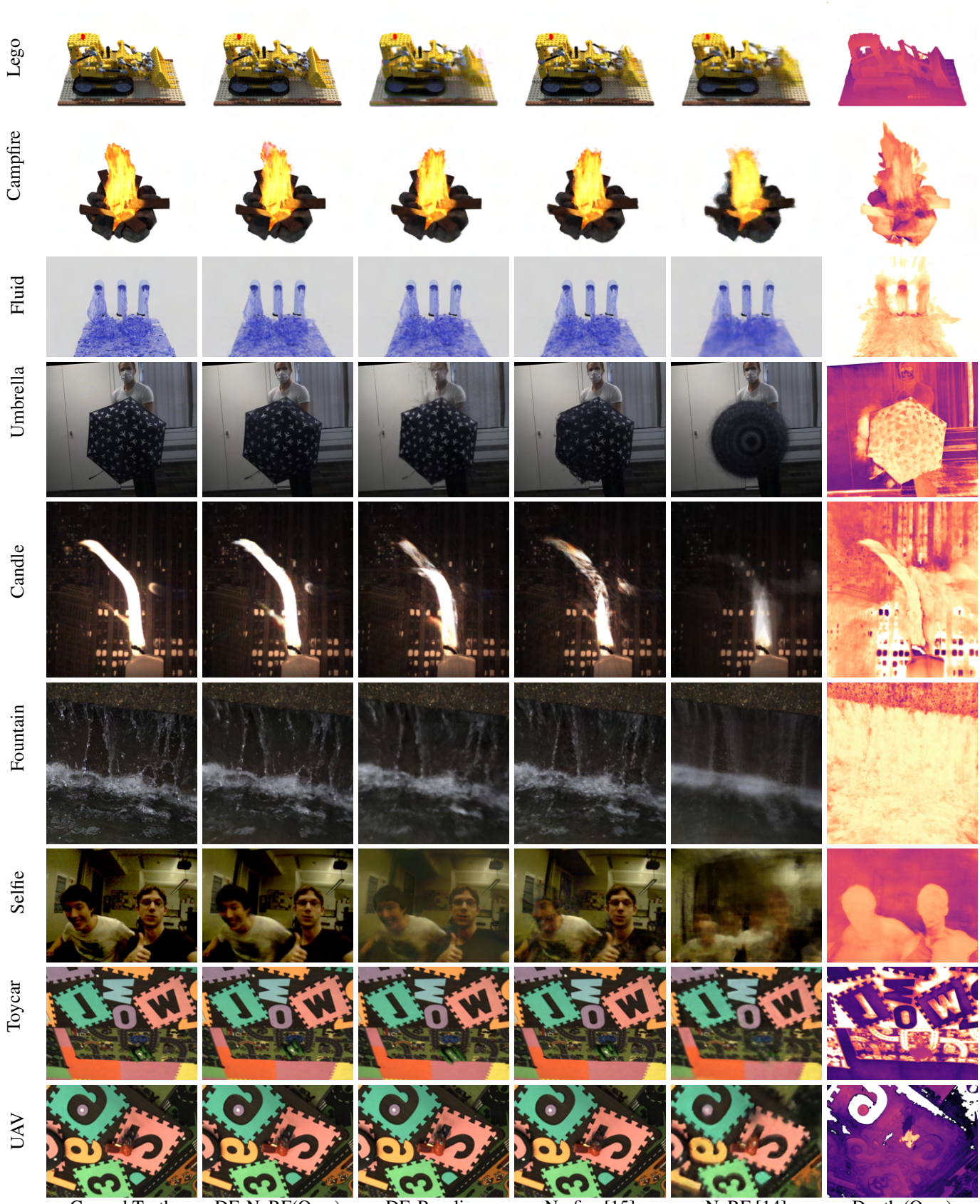

Table 3. Qualitative comparisons of our method and baselines on synthetic and real-world datasets.

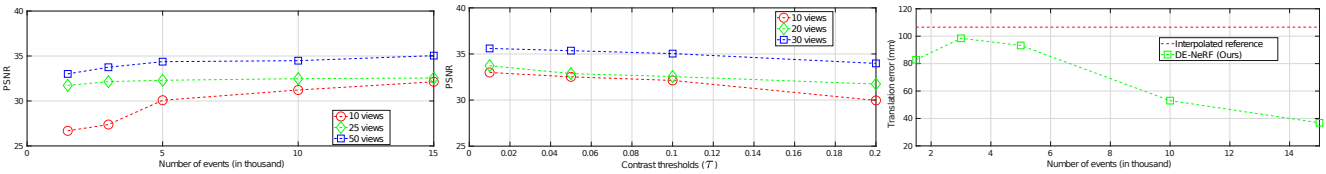

Figure 6. The behaviour analysis of our method. Novel view synthesis quality vs. RGB views and number of views (left) and even contrast threshold (τ) for sensitivity measure (middle). The pose error measure in absolute translation error vs. number of events used (right).

Number of Views	10		25		50	
Method	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
Nerfies[15]	18.51	0.351	22.51	0.147	25.97	0.089
DE-Baseline	22.21	0.113	23.95	0.101	27.12	0.093
Ours (no void)	28.16	0.076	29.82	0.044	33.25	0.038
Ours	28.89	0.078	29.91	0.042	33.41	0.040
Ours + AS	28.85	0.102	31.83	0.039	34.60	0.038
Ours + AS + PR	32.13	0.046	32.55	0.037	35.04	0.034

Table 4. **Ablation Study.** We investigate the effectiveness of void sampling, active sampling (AS) as well as pose refinement (PR). All the proposed components contribute meaningfully to our method. The gain of our method without additional components comes from event integration alone performed using (1).

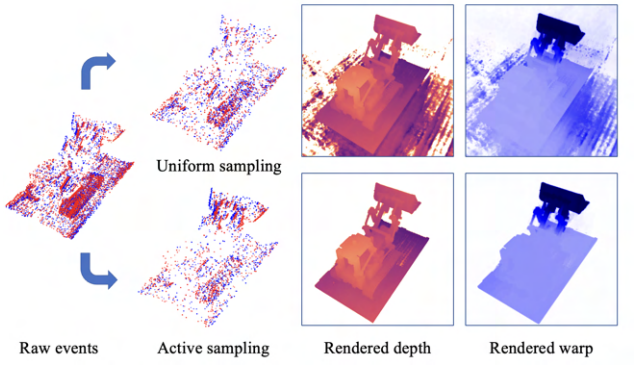

Figure 7. Uniform vs. active sampling techniques. Our active sampling method uses more events from deformable regions.

-	E2VID[20]+Nerfies[15]		E2VID+Hyper[16]		E2VID	
Dataset	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
Lego	17.12	0.46	16.40	0.50	15.17	0.38
Umbrella	25.05	0.44	24.93	0.46	25.92	0.15
Selfie	18.36	0.39	17.95	0.40	16.95	0.42

Table 5. **Events-to-frame based method Comparison.** We report results using learning based events-to-frame method E2VID[20]. It can directly synthesize novel view frame using only events. However, as depicted in the third column, the synthesized quality is deficient.

tensive experiments on diverse real and synthetic datasets revealed very exciting results with significant performance gain, both quantitatively and qualitatively. The success of

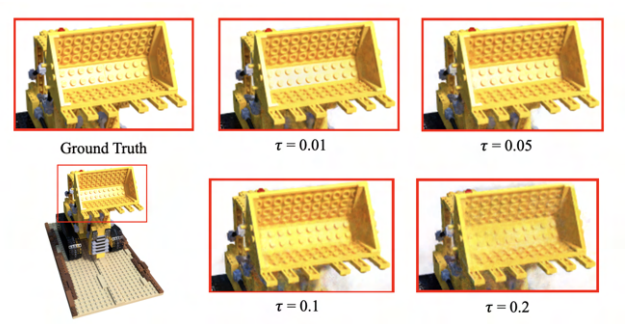

Figure 8. Effect of contrast thresholds on view synthesis. As expected, more sensitive event cameras lead to better representations.

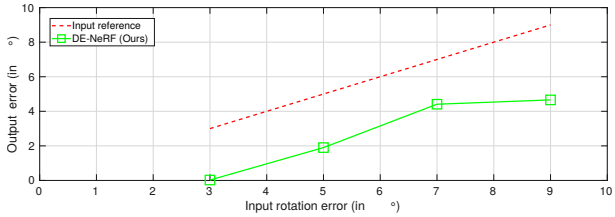

Figure 9. PoseNet robustness against injected rotation noise.

the proposed method must also be credited to the recent advancements in radiance field modeling. This is particularly the case because the integration of event cameras in the radiance field is in fact very natural. This allowed us to quickly establish a baseline and improve it using the techniques proposed in this paper. Our method opens new avenues for the 3D visual modeling of fast-moving cameras and deforming scenes, in a relatively simple manner.

Limitations: For monochromatic events, our method occasionally generates color artifacts. Our method benefits insignificantly in very complex scenes that largely violate the assumed deformation model. This can be seen with the Fluids dataset. We believe this limitation can be addressed by more sophisticated non-rigid priors for complex scenes.

Acknowledgements: Research is partly funded by VIVO Collaboration Project on Real-time scene reconstruction.

References

[1] Alexis Baudron, Zihao W Wang, Oliver Cossairt, and Aggelos K Katsaggelos. E3d: event-based 3d shape reconstruction. *arXiv preprint arXiv:2012.05214*, 2020.

[2] Christian Brandli, Lorenz Muller, and Tobi Delbruck. Real-time, high-speed video decompression using a frame-and event-based davis sensor. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 686–689. IEEE, 2014.

[3] Samuel Bryner, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 325–331. IEEE, 2019.

[4] Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. Evimo2: An event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv preprint arXiv:2205.03467*, 2022.

[5] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021.

[6] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems*, 2022.

[7] 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.

[8] 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*, pages 837–847, 2023.

[9] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. 2014.

[10] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 349–364. Springer, 2016.

[11] Simon Klenk, Lukas Koestler, Davide Scaramuzza, and Daniel Cremers. E-nerf: Neural radiance fields from a moving event camera. *IEEE Robotics and Automation Letters*, 2023.

[12] 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.

[13] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.

[14] 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.

[15] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[16] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), dec 2021.

[17] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[18] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time. *International Journal of Computer Vision*, 126(12):1394–1414, 2018.

[19] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.

[20] 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.

[21] Viktor Rudnev, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. Eventnerf: Neural radiance fields from a single colour event camera. *arXiv preprint arXiv:2206.11896*, 2022.

[22] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. Eventhands: Real-time neural 3d hand pose estimation from an event stream. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12385–12395, 2021.

[23] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12371–12381, 2022.

[24] Cedric Scheerlinck, Henri Rebecq, Timo Stoffregen, Nick Barnes, Robert Mahony, and Davide Scaramuzza. Ced: Color event camera dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[25] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[26] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 12959–12970, 2021.

[27] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. TimeLens: Event-based video frame interpolation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[28] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[29] Ziyun Wang, Kenneth Chaney, and Kostas Daniilidis. Evac3d: From event-based apparent contours to 3d models via continuous visual hulls. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 284–299. Springer, 2022.

[30] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16210–16220, 2022.

[31] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13296–13306, June 2022.

[32] Yuxuan Xue, Haolong Li, Stefan Leutenegger, and Jörg Stückler. Event-based non-rigid reconstruction from contours. *arXiv preprint arXiv:2210.06270*, 2022.

[33] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild, 2023.

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[35] Zerong Zheng, Tao Yu, Qionghai Dai, and Yebin Liu. Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1429–1439, June 2021.

[36] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 235–251, 2018.

[37] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics*, 37(5):1433–1450, 2021.

[38] 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.