Continuous Pose for Monocular Cameras in Neural Implicit Representation

Qi Ma[1,2] Danda Pani Paudel[2] Ajad Chhatkuli[1] Luc Van Gool[1,2]
[1]Computer Vision Lab, ETH Zurich [2]INSAIT, Sofia University

Abstract

In this paper, we showcase the effectiveness of optimizing monocular camera poses as a continuous function of time. The camera poses are represented using an implicit neural function which maps the given time to the corresponding camera pose. The mapped camera poses are then used for the downstream tasks where joint camera pose optimization is also required. While doing so, the network parameters – that implicitly represent camera poses – are optimized. We exploit the proposed method in four diverse experimental settings, namely, (1) NeRF from noisy poses; (2) NeRF from asynchronous Events; (3) Visual Simultaneous Localization and Mapping (vSLAM); and (4) vSLAM with IMUs. In all four settings, the proposed method performs significantly better than the compared baselines and the state-of-the-art methods. Additionally, using the assumption of continuous motion, changes in pose may actually live in a manifold that has lower than 6 degrees of freedom (DOF) is realized. We call this low DOF motion representation as the intrinsic motion *and use the approach in vSLAM settings, showing impressive camera tracking performance. We release our code at: https://github.com/qimaqi/Continuous-Pose-in-NeRF.*

1. Introduction

The concept of motion, the change of position and orientation of an object in its surroundings, is fundamentally continuous in nature. This continuity is evident in the ways we achieve, perceive and measure motion, with velocity and acceleration being the most common measures for both linear and angular motion. This idea of continuity is also true for the 3D poses of navigating cameras. Often the camera motion needs to be estimated from its measurements – also known as the camera localization problem. In most common settings, the inputs are RGB-only frames, depth frames, asynchronous event streams, or a combination thereof. In some cases, these measurements are augmented by Inertial Measurement Unit (IMU) outputs, which measure a change in pose directly. In all those settings, the camera motion is estimated via some optimization technique that searches $SE(3)$ pose parameters. While doing so, existing techniques choose to optimize a discrete set of $SE(3)$ parameters, ignoring the inter-frame continuity of camera poses. This choice can be primarily attributed to the otherwise difficulty in optimization.

While handling high-frequency IMUs or asynchronous events in common practice, pose optimization at every measurement time is avoided, for computational reasons. Instead, the measurements between two arbitrarily chosen keyframes are accumulated before utilizing them. Then the poses are optimized only for those keyframes. We argue that this raises three major concerns: (i) inaccurate accumulation of intermediate measurements; (ii) loss of fine-grained motion details; (iii) lack of the continuous motion prior.

In order to address these concerns, we represent and optimize the pose of a moving camera as a continuous function of time. Unlike classical state estimation method [2, 12, 35] which models continuous pose with Gaussian Process or B-spline, our neural pose function can be easily optimized jointly with other task-specific implicit neural representation (INR) [30, 32, 39]. More precisely, for translation $\mathsf{v} \in \mathbb{R}^3$ and rotation $\mathsf{R}(\mathsf{q}) \in SO(3)$ parameterized by quaternions $\mathsf{q} \in \mathbb{R}^4$ with $||\mathsf{q}|| = 1$, the continuous pose of the monocular camera is given by,

$$[\mathsf{q}; \mathsf{v}] = f_\theta(t), \tag{1}$$

where $f_\theta(.)$ is the contineous neural function parameterized by θ that maps the time $t \in \mathbb{R}$ to the pose in $SE(3)$. While being simple, this representation has numerous benefits including ease of optimization and its cosmopolitan applicability. Some example applications are illustrated in Figure 1. In the following, we further discuss how our simple approach addresses the previously raised concerns.

No error due to measurement accumulation: High frequency or asynchronous measurements can be utilized directly without accumulation, integral, or rounding. We infer the camera pose precisely at the measurement time. For example, in the case of an event camera, each asynchronous event's pose is inferred precisely at the event time. Similarly, in the case of IMUs, no motion integration before supervision is required. These abilities protect our approach against error injection due to any form of accumulation.

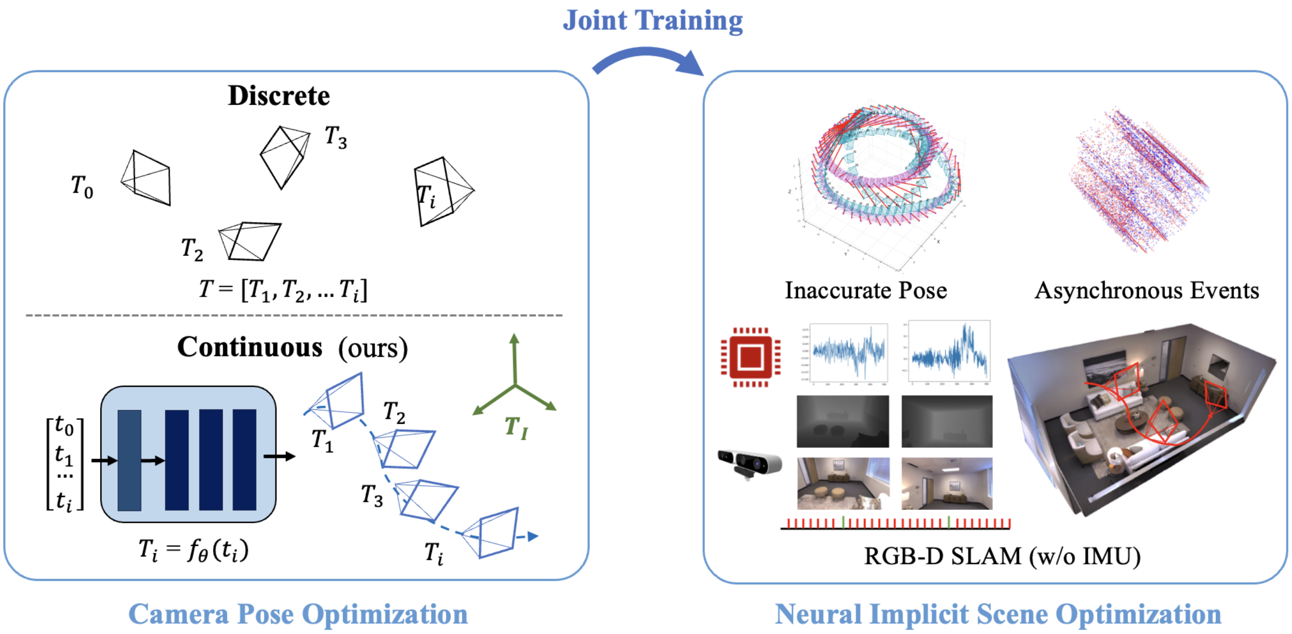


Figure 1. We showcase the benefits of optimizing the poses as a continuous function of time in diverse settings. We conduct exhaustive experiments on (a) rectifying inaccurate poses in RGB-only settings; (b) utilizing the asynchronous stream of events, (c) performing vSLAM in RGB-D camera settings; (d) integrating high-frequency IMUs in vSLAM. All experiments use neural functions for both camera poses and scene representations. Additionaly we exploit low dof motion representation in intrinsic motion frame T_I.

Fine-grained motion details: By virtue of the continuous representation, temporally fine-grained details of the pose can be captured. This is particularly interesting with high-frequency IMUs or asynchronous event cameras. Our approach allows for the recovery of the pose at the very moment of measurement, which otherwise often is an ill-posed problem and could only be interpolated with an assumed smoothness and order.

Continuous motion prior: The inductive bias of continuous monocular camera motion is meaningfully injected by the proposed method. This resulted in very encouraging results in our experiments. In particular, while denoising the inaccurate camera poses and during the vSLAM experiments, the benefits were evident under the standard settings of BARF [27] and NICE-SLAM [63], respectively. It is important to note that our representation offers first- and second-order derivatives via auto-differentiation of the neural network. Consequently, quantities such as velocity and acceleration do not require additional care. Thus the fusion of IMU measurements is natural and straightforward.

In addition to the above, we further show the utility of the neural pose in order to optimize the continuous pose by decomposing each change in pose into a slowly changing reference and a low DOF motion. We define this as the *intrinsic motion*. In our experiments we observed that our continuous pose representation improves the tracking performance significantly in the vSLAM tasks. This can be primarily attributed to the reasons mentioned above, which serve to facilitate the optimization process. Inspired by the fact that actual motion always possesses a lower degree of freedom, we define the intrinsic motion frame as a coordinate system that can express the camera motion with the lowest dimensional manifold. For example: Rotational motion around a fixed axis can be expressed in the coordinate system aligned with the rotational axis with only one degree of freedom. A natural observation is that the relative motion with respect to intrinsic motion is usually sparse, moreover, the continuous motion tends to share the same intrinsic motion frame which can be well modeled as a continuous function of time. By exploiting it we decompose the camera relative motion with a low-dimensional intrinsic motion $[\mathsf{R}_I, \mathsf{v}_I]$ and the rigid transformation from camera frame to the intrinsic motion frame $[\mathsf{R}_o, \mathsf{v}_o]$ as follows:

$$[\mathsf{R}, \mathsf{v}] = [\mathsf{R}_o, \mathsf{v}_o][\mathsf{R}_I, \mathsf{v}_I], \tag{2}$$

Our major contributions can be summarized as follows:

- We propose a simple yet effective way to represent the monocular camera motion via a neural function of time that can be optimized efficiently together with implicit neural representations.
- We demonstrate the utility of the proposed representation in four diverse applications with different camera setups, including IMUs and moving event cameras.

- Through exhaustive experiments, we demonstrate clear benefits of the proposed representation over the existing alternatives and classical method. These benefits include ease of optimization, widespread use for different camera and sensor types, and notable performance gain with no additional effort.
- We further improve the full 6-DOF pose of monocular camera by exploiting the sparsity of the intrinsic motion, which fits neatly into the proposed framework of continuous neural pose. The final pose thus obtained shows remarkable improvement over the conventional baselines.

2. Related work

2.1. Camera Poses in NERF

NERF [32] consists of joint optimization of the surface density and the rendered color given the images with known camera rays. Consequently NERF models are highly sensitive to camera pose errors [8, 27, 29, 55, 58, 59]. Recently several works have tackled the pose error by jointly optimizing poses with the radiance field. [6, 7, 19, 27] optimizes camera poses in bundle adjustment fashion in order to solve the same issue. While these methods use the smooth pose prior, the poses are still optimized as discrete variables. On the other spectrum [55] optimizes noisy poses for sparse camera views with the radiance fields opting for a different class of applications. [3] enforces the inter-frame consistency by incorporating monocular depth prior.

2.2. Camera Poses with IMUs

The inertial measurement unit (IMU) serves as a scene-independent sensor that is the ideal complement to cameras in order to achieve robustness in low texture, high speed, and HDR scenarios. Fusing visual information and IMU tightly [48] to estimate pose as discrete states is proposed first by MSCKF [34] (an extended Kalman Filter (EKF)), [25] further improves it with keyframes and bundle adjustment. [11, 17, 41, 56] improve in robustness compared to feature matching by using the direct photometric error. [5] propose fast and accurate IMU initialization based on MAP estimation. Recent research has also focused on integrating IMU and visual priors with neural network, *e.g.*, the camera pose is implicitly used for image deblurring [37] or video stabilization [49]. [14] proposes neural inertial localization with IMU alone for indoor scenes.

2.3. Camera Poses in Dense SLAM

Visual SLAM [10, 23, 43] is a key 3D vision application where an agent camera is localized simultaneously while building the map using visual information. We again focus on methods in the context of radiance fields [1, 44, 53, 63, 64]. IMAP [53] is a recent seminal work which works on RGBD images to optimize an implicit scene representation with a single MLP network. It optimizes the camera pose while representing them as discrete sets of parameters for the keyframes. NICE-SLAM [63] improves on it by using 3D voxel features along with corresponding 2D image features thus providing a better scene representation. Indeed most approaches [26, 44, 47, 64] focus on improving the scene representation for better localization and mapping or with RGB-only input.

2.4. Camera Poses in Event Cameras

Unlike standard frame-based camera imaging, event cameras provide image signals as asynchronous events in microsecond intervals [22]. Thus, it forms the perfect use case for a continuous time representation of camera poses. Similar to NERF-less SLAM [10], this is traditionally done using variations of Kalman Filter with motion models [22, 33]. A recent work [62] represents camera tracking as a function of time but uses a Levenberg-Marquardt optimization directly on the sets of poses without intermediate representation. Recently, there have been efforts to use event-based radiance fields in the neural network [18, 24, 28, 46]. However, camera pose optimization as a function of time is still not fully explored in the radiance field literature with events.

2.5. Continuous Pose representation

While discrete-time representations are commonly employed in Simultaneous Localization and Mapping (SLAM) tasks, they face challenges when integrating data from high-frequency sensors like Inertial Measurement Units (IMUs) and asynchronous events. [12] address this issue by proposing representing the continuous-time state using temporal basis functions such as B-spline basis.[2] model the continuous state using Gaussian processes, defining continuous-time priors through covariance functions.[40] leverage cumulative cubic B-splines to mitigate rolling-shutter artifacts. Notably, spline-based continuous-time trajectory representations have found application in laser-based SLAM methods [21, 38].

3. Time-to-Pose Mapping Network

3.1. Architecture of the Proposed PoseNet

In order to learn time-to-pose mapping, we use 8-layer MLP parameterized by $f(\theta_p)$ with ReLU activation functions and 256-dimensional hidden units, which we refer to as pose-network (PoseNet). PoseNet first embeds the time variable into high-dimensional space using sinusoidal harmonic functions [32]. The outputs of this network are $[\mathsf{v}, \mathsf{q}]$: translation vector $\mathsf{v} \in \mathbb{R}^3$ and the rotation represented by a quaternion $\mathsf{q} \in \mathbb{R}^4$. Finally, we use the tanh activation in the last layer to map output to the range $[-1, 1]$, and normalize it as a unit quaternion. We study different embedding dimensions and architectures in the context of NeRF from the inaccurate pose, which is reported in Tables 3. The best-performing embedding and architecture, in these experiments, are then used for the other applications. Additional information concerning network size and computational details is provided in the supplementary material.

3.2. Implementation Variances across Applications

The simplicity of PoseNet allows us to use it in diverse applications in a plug-and-play manner. In all applications that we report in the following sections, we optimize the PoseNet parameters θ_p as a surrogate of the direct pose optimization. We denote the network parameters for the INR of the scene as θ_s. In NeRF from inaccurate poses, the objective is to minimize the radiance field loss [27, 32] given N images and corresponding timestamp t_i for image i:

$$\min_{\theta_s,\theta_p} \sum_{i=1}^{N} \|\mathcal{I}_i - g\left(\theta_s, f(\theta_p, t_i)\right)\|. \tag{3}$$

$g(\theta_s, T_i)$ represents the mapping from the camera pose T_i to the RGB value, including ray composition and radiance field model. In case of NeRF with asynchronous events [46], N refers to the number of the sampled events. Note that in both cases, we output the predicted transformation and compose it with the initial pose: $T_i = T_{init_i} \circ T_{refine_i}$. The refined transformation is obtained as $T_{refine_i} = P(f(\theta_p, t_i))$, $P(.)$ being the vector to rigid transformation conversion operator.

In the task of Dense-SLAM tracking, for each tracking iteration we optimize PoseNet with the following objective:

$$\min_{\theta_p} \sum_{i=1}^{M} (\mathcal{L}_g(D_i, P(f(\theta_p, t_i))) + \lambda_p \mathcal{L}_p(I_i, P(f(\theta_p, t_i)))). \tag{4}$$

We use the same geometric loss $\mathcal{L}_g$ and photometric loss $\mathcal{L}_p$ as in NICE-SLAM [63]. D_i, I_i represent depth and RGB measurements for M sampled pixels respectively, obtained via volume rendering.

3.3. Intrinsic Motion Frame

Within the neural dense SLAM application, we additionally introduce intrinsic motion frame in order to improve tracking within a low-dimensional manifold. This is accomplished through motion decomposition and enforcing minimal DOF. More specifically, we use two PoseNet $f_o(\theta_{p_o})$, $f_I(\theta_{p_I})$ in order to model the intrinsic motion $T_o = [\mathsf{R}_o, \mathsf{v}_o]$, $T_I = [\mathsf{R}_I, \mathsf{v}_I]$, such that $T = T_o \circ T_I$. Here, T_o is the transformation to the *intrinsic frame* or in short, intrinsic transform. T_I then denotes the intrinsic motion. Therefore we can rewrite Eq (4) as:

$$\begin{aligned} \min_{\theta_{p_o},\theta_{p_I}} \sum_{i=1}^{M} & (\mathcal{L}_g(D_j, f_o(\theta_{p_o}, t_i) \circ f_I(\theta_{p_I}, t_i)) \\ & + \mathcal{L}_p\left(I_j, f_o(\theta_{p_o}, t_i) \circ f_I(\theta_{p_I}, t_i)\right) \\ & + \mathcal{L}_{dof}(f_I(\theta_{p_I}, t_i)) + \mathcal{L}_o(f_o(\theta_{p_o}, t_i)). \end{aligned} \tag{5}$$

Note that the operator P should be included for absolute correctness in the function compositions in Eq (5). The DOF loss $\mathcal{L}_{dof}$ is computed as follows:

- **Step1**: Obtain $[\mathsf{R}_I, \mathsf{v}_I]$ from intrinsic motion PoseNet f_I
- **Step2**: Convert rotation matrix R_I to Euler angles $\alpha_I \in \mathbb{R}^3$, normalize with angle of view γ, $\hat{\alpha_I} = 2\alpha_I/\gamma$
- **Step3**: Normalize translation vector with $\hat{v_I} = v_I/\|v_I\|$.
- **Step4**: DOF Loss $\mathcal{L}_{dof}$ = $\|[\hat{\alpha_I}, \hat{v_I}]\|_0$.

We relax the ℓ_0 norm to ℓ_1 norm for optimization. In steps 2 and 3, normalization also serves to balance translation and rotation components during optimization. We employ view angle normalization with the assumption that the angle between two relative views in vSLAM tasks is always smaller than half of the viewing angle. To handle the cases where unconstrained intrinsic motion tends move to infinity in cases of small rotation, we introduce an additional $\mathcal{L}1$ regularization term for the translation $\mathcal{L}_o = |v_o|$.

3.4. IMU fusion

Up to our knowledge we are the first to integrate IMU data in NeRF + SLAM setting. The IMU fusion is straightforward in PoseNet taking advantage of the auto-differentiation of the neural network. We propose two different IMU fusion methods with details as follows:

Loose coupling. Given 3-axis angular velocity measurement from gyroscope $\hat{\omega} = (\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z)$ we get the time step from frequency $\triangle t = \frac{1}{f}$. We can express the rotation angle to be $\triangle t\|\hat{\omega}\|$ around axis $\frac{\hat{\omega}}{\|\hat{\omega}\|}$ [50]. This instantaneous rotation from the local sensor between previous and current timestamp can be represented as follows:

$$\mathsf{q}_{\triangle} = \mathsf{q}\left(\triangle t\|\hat{\omega}\|, \frac{\hat{\omega}}{\|\hat{\omega}\|}\right). \tag{6}$$

By continuously integrating the measurements we can get the rotation estimation at time t_i with respect to t_{j-1} from gyroscope: $\mathsf{q'}_{t_i} = \mathsf{q}^{(t_{j-1})}\mathsf{q}_{\triangle}$. We add ℓ_1 loss $\mathcal{L}_{loose} = |q_{t_i} - q'_{t_i}|$ into Eq 4, where q_{g_j} integrate all gyroscope measurements from timestamps t_{j-1} to t_j.

Tight coupling. However, simply integrating IMU information leads to large drift and noise over time. As an immediate consequence of our continuous pose representation over time, we can directly fuse the angular velocity using the quaternion derivative [50]:

$$\dot{\mathsf{q}} = \frac{1}{2}\Omega(\hat{\omega}). \tag{7}$$

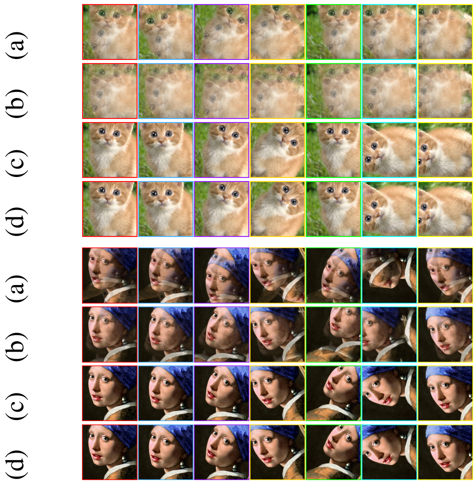


Figure 2. **Patch Reconstruction** Color-coded patch correspond to Fig 3.Note that patch 2D rigid motion exhibits continuity over time (left to right)

Method	Cat		
	CE(pixel) ↓	PSNR ↑	SR ↑
BARF[2]	13.55	27.61	30%
B-spline	35.14	21.95	0%
Ours	**0.01**	**37.00**	**100%**
	Girl		
BARF[2]	29.94	22.09	15%
B-Spline	39.08	19.42	10%
Ours	**6.92**	**32.40**	**95%**

Table 1. **Image alignment experiment.** Results of average 20 sampled 2D rigid motion, CE refer to Corner Error and SR refer to successful rate.

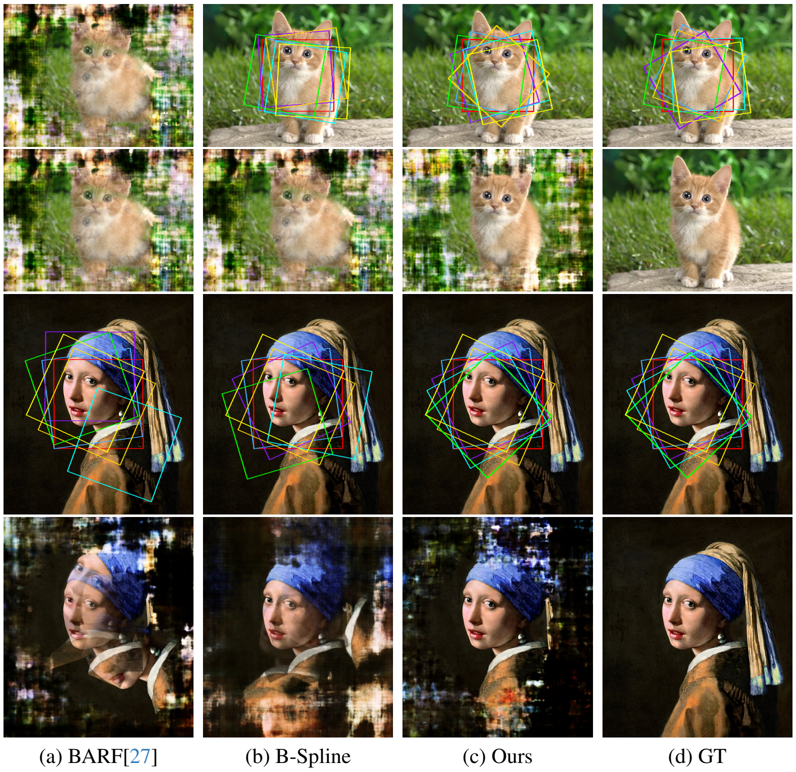

(a) BARF[27] (b) B-Spline (c) Ours (d) GT

Figure 3. **Qualitative results of 2D planar Alignment.** We report the results of planar image alignment. Given input as ground truth (d) shown in Fig. 2, the goal is to find the 2D rigid transformation for each patch and optimize the entire neural image. Our method optimizes for accurate alignment and high-fidelity image reconstruction, while baselines fail due to local minima.

Thus we can supervise PoseNet by constraining the jacobian with the measured angular velocity. We use ℓ_1 loss as $\mathcal{L}_{tight} = |\dot{\mathsf{q}} - \frac{1}{2}\Omega(\hat{\omega})|$ and jointly optimize it with the tracking target function in Eq 4.

It is noteworthy that, in the aforementioned equation, our PoseNet outputs pose with respect to the body frame rather than the camera frame. Further details regarding the coordinate change can be found in the supplementary materials, along with an explanation of how acceleration is utilized.

4. Experiments

4.1. NeRF from Inaccurate Poses

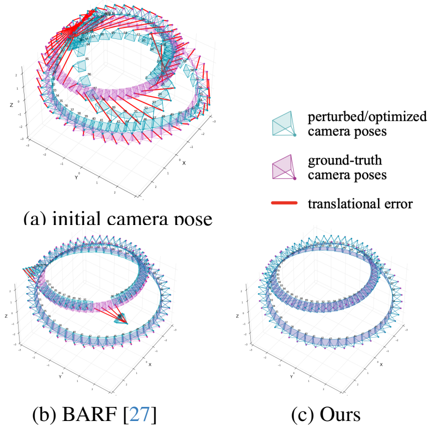
We validate the effectiveness of our proposed method through 2D planar image alignment experiments and 3D scene experiments similar to BARF [27]. During this process, BARF refines a discrete set of inaccurate camera poses while our method leverages the continuous pose information and is therefore less prone to local minima.

4.1.1 Planar Image Alignment (2D)

We choose the same images as [7, 27] as shown in Fig 3. To obtain a continuous rigid transformation, we initially randomly sample 10 data points from $T \in SE(2)$, we then interpolate a cubic spline along each dimension. Finally, we interpolate on 7 uniformly spaced points at previous time instants. As a result, the rigid transformation demonstrates temporal correlation, as illustrated in Fig 2. The initialized pose is identity with respect to center crop.

Experimental settings. We compare our method against BARF [27] and BaRF with B-spline. For the latter, we introduce continuity by resetting the learned $T \in SE(2)$ for every 100 steps using B-spline interpolation. The learning rate is $1e-3$ for translation and $2e-4$ for rotation. For the B-Spline method we report with 5 knots placed time-wise uniformly with degree $= 3$.

Results. The results are visualized in Fig 2,3. The alignment performance of BARF suffers from local minima, resulting in sub-optimal performance. Experiments are deemed successful if the corner error is below 1 pixel. Although some patches correctly learn the transformation,



(a) initial camera pose

(b) BARF [27]

(c) Ours

Figure 4. We introduce continuous errors on the camera trajectories and perform pose refinement in the NeRF setting. (a) Initial pose error; (b) results obtained using BARF [27] that uses a discrete set of poses; (c) results obtained using our continuous pose representation.

Scene	Rotation ↓		Translation ↓		PSNR ↑		SSIM ↑		LPIPS ↓	
	BARF	ours	BARF	ours	BARF	ours	BARF	ours	BARF	ours
Fern	0.199	0.181	0.196	0.181	21.01	21.08	0.62	0.63	0.33	0.31
Fern/2	0.344	0.331	0.195	0.173	19.72	19.74	0.53	0.53	0.33	0.32
Fern/4	0.289	0.264	0.212	0.215	19.65	21.33	0.54	0.63	0.33	0.32
Fortress	0.444	0.360	0.369	0.283	23.17	22.17	0.48	0.41	0.12	0.12
Fortress/2 *	6.507	0.574	3.545	0.418	14.86	20.00	0.35	0.33	0.40	0.17
Fortress/4	0.607	0.630	0.583	0.629	20.71	20.72	0.38	0.40	0.17	0.20
Orchids	0.719	0.645	0.390	0.364	13.22	14.46	0.17	0.24	0.35	0.30
Orchids/2	0.809	0.730	0.387	0.375	12.60	13.50	0.15	0.19	0.37	0.35
Orchids/4 *	92.176	0.865	46.772	0.388	11.07	12.64	0.18	0.16	0.97	0.49
Room	0.288	0.106	0.245	0.101	21.78	25.32	0.79	0.88	0.14	0.10
Room/2	0.329	0.274	0.284	0.172	21.20	21.27	0.77	0.78	0.13	0.13
Room/4 *	118.58	0.403	76.14	0.550	11,00	22.76	0.42	0.80	0.89	0.17
Average	18.44 (0.448)	**0.446** **(0.391)**	10.777 (0.318)	**0.320** **(0.276)**	17.50 (19.23)	**19.58** **(19.95)**	0.448 (0.492)	**0.498** **(0.520)**	0.378 (0.252)	**0.248** **(0.238)**

Table 2. **Real data with unknown pose.** Our PoseNet compared to BARF [27] for the real dataset, simulating different camera moving speeds. Whenever BARF diverges and provides very inaccurate results, we consider them failures and denote them as *. The average across all experiments is provided for all (and averaged only when BARF succeeds). In addition to the 12/12 (Ours) vs. 9/12 (BARF) success rate, PoseNet performs better than BARF also in cases when BARF succeeds.

Method	RE ↓	TE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LE, C	13.62	48.05	9.79	0.59	0.56
Sinusoidal(2), C	3.70	15.84	14.05	0.66	0.20
Sinusoidal(5), C,	**0.07**	<u>0.32</u>	<u>27.25</u>	<u>0.91</u>	**0.05**
Sinusoida(10), C	<u>0.18</u>	0.88	24.88	0.88	<u>0.06</u>
Sinusoidal(2), D	2.86	10.53	15.97	0.69	0.14
Sinusoidal(5), D	**0.07**	**0.28**	**27.30**	**0.92**	<u>0.06</u>
Sinusoidal(10), D	**0.07**	**0.28**	27.20	<u>0.91</u>	**0.05**
Sigmoid, D	14.31	37.21	11.28	0.67	0.55
Sinusoidal(10) c2f, D	14.74	49.07	9.78	0.60	0.60

Table 3. **Ablation study.** We investigate the effectiveness of our PoseNet with diverse architecture. LE refers to linear encoder and C, D refer to coupled and decoupled representations. RE, TE refer to rotational and translation error. The best and second-best results are in bold and underlined.

they do not effectively contribute to neighbouring patches. Merely introducing B-spline directly does not work, as it can over-smooth or under-smooth the poses, whereas our proposed method successfully captures all rigid transformations resulting in high-fidelity neural image. Furthermore as demonstrated in Table 1 our method performs consistently well across 20 different trajectories.

4.1.2 Synthetic and Real NeRF (3D)

We explore the more challenging problem of learning 3D Neural Radiance Field with inaccurate poses. For the synthetic data, we render Lego [32] with a circular movement as shown in Figure 4. The simulated camera orbits the Lego model in the xy-plane, moving up and down at a constant speed in the z-direction.

Experimental settings. Similar to the 2D experiment, we introduce temporal correlation between neighboring $SE(3)$ disturbances with interpolation. We use spherical linear interpolation for rotation. The introduced error corresponds to 55° in rotation and 110% in translation. For real data, we use the Fern, Fortress, Orchids, and Room datasets in LLFF [31], since these sequences allow us to perform experiments with varying numbers of images, thus simulating fast-moving cameras. Unlike in the synthetic case, *we do not use any pose initialization* in the real data experiments. Following [27] we report the MSE distance and rotational angle after alignment using Procrustes analysis for registration evaluation and PSNR, SSIM [57] and LPIPS [61] to evaluate view synthesis quality.

Results. We report our experimental results in Table 3 and Table 2, for synthetic and real data, respectively. In Table 2, the proposed continuous pose representation clearly offers better results than the discrete BARF. Ablation experiments further illustrate that the rotation and translation decoupled representation, i.e., two MLPs instead of one, performs better, offering the best results with the embedding frequency bands $F = 5$. In real data with completely unknown camera poses, PoseNet performs significantly better than BARF. These results are reported in Table 2, where dataset/n refers to $1/n^{th}$ fraction of uniformly downsampled cases. It can be seen that PoseNet successfully handles all three failure cases of BARF. This is particularly evident when only sparse images are available. At the same time, even when BARF succeeds, PoseNet performs significantly better than BARF. More results and experiments using B-Spline can be checked in supplementary material.

	TE ↓		PSNR ↑		SSIM ↑		LPIPS ↓	
Num	without	ours	without	ours	without	ours	without	ours
Chair								
20	3.66	**1.74**	26.36	**26.58**	0.89	**0.91**	0.19	**0.15**
10	15.63	**3.38**	22.48	**25.02**	0.81	**0.86**	0.34	**0.18**
6	59.31	**22.68**	21.45	**22.06**	0.70	**0.80**	0.57	**0.34**
Hotdog								
20	3.66	**2.42**	23.59	**25.64**	0.90	**0.92**	0.14	**0.10**
10	15.63	**4.87**	21.85	**23.15**	0.85	**0.87**	0.20	**0.16**
6	59.31	**6.70**	21.06	**22.03**	0.79	**0.85**	0.34	**0.18**

Table 4. **Interpolation error experiments.** We improve the EventNeRF [46] using PoseNet. A small number of sparsely known poses are used to estimate the poses in between. PoseNet improves EventNeRF significantly in all six experimental setups.

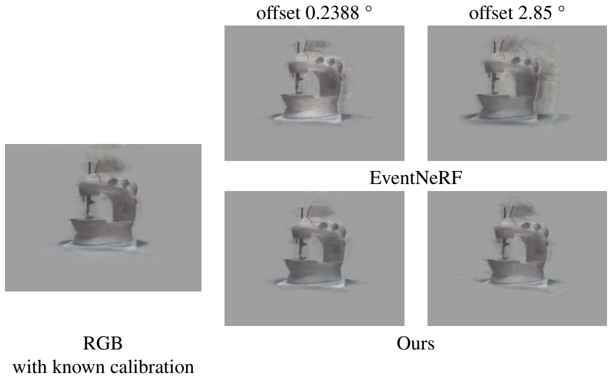


Figure 5. **With and without calibration experiments.** We investigate the effectiveness of our method under different deviations from the actual rotational axis. Our method can successfully reposition the object back to the center without additional calibration.

4.2. Continuous Pose for Asynchronous Events

By virtue of continuous pose representation, handling asynchronous event streams acquired by event cameras becomes natural. Hence, we use our PoseNet to learn the radiance field-based 3D scene representation from only colour event streams. This experimental setup is similar to recent work EventNeRF [46]. Note that EventNeRF accumulates asynchronous events to high-frequency synchronous event frames. The poses of each of those event frames are then assumed to be known. We argue that these assumptions limit the potential of the event cameras which come from their asynchronous nature. Therefore, we query for the pose of every event precisely at their trigger times. We conduct two experiments to address two practical issues of using events in EventNeRF setup using both synthetic and real datasets.

4.2.1 Unknown continuous pose for single event

Events are triggered asynchronously, and in practice where there is no precisely measured control available such as with a turntable [46] or a motorized linear slider [42], event pose can only be interpolated from measured discrete poses (from Vicon or Colmap [15]). However, this introduces interpolation errors.

Experimental settings. For synthetic data, we use *chair* and *hotdog* sequences from [46]. The events are simulated using the model in [45]. While EventNeRF performs interpolation, our method jointly learns intermediate poses as a continuous function of time.

Results. In Table 4, we reveal that integration of our PoseNet significantly enhances the overall performance with a notable reduction in translation errors and better scene reconstruction. More visual results can be found in supplementary material.

4.2.2 Unknown calibration in practice

EventNeRF [46] uses turntable to achieve stable and consistent object rotation speed. This setup also requires the actual rotational axis. Therefore, an additional checkerboard-based calibration technique, to estimate the axis offset, is also proposed in [46].

Experimental setting. For real cases, we use *sewing machine* datasets, which hold difficulties in reconstructing thin structures, view-dependent effects, and colored texts.

Results. We show that when PoseNet is used, additional calibration may not be required. The qualitative results of these experiments are shown in Figure 5. We demonstrate that when some offset is introduced, the 3D object deviates from the center for EventNeRF, while our method can reduce artifacts, learn the offset angle, and reposition the object back to the image center.

4.3. Visual SLAM with Depth and IMUs

While the previously discussed tasks are offline, vSLAM is an online method with different considerations. In this application we approach the problem as incremental SLAM. For each incoming frame, our objective is to determine its transformation with respect to the last frame $T_{relative}$. Similar to NICE-SLAM [63] we maintain a list of all optimal relative poses. It is trivial to solve the forgetting issue by retraining our PoseNet with such a list.

Experimental settings. We report the tracking results of our method compared with the standard NICE-SLAM. We report results of intrinsic motion on Replica [51], Scannet [9] and TUM-RGBD[52]. Note that during tracking we assume intrinsic motion reference slowly changes over time and only optimize f_o for every keyframe, with a frequency set to 10 for our experiments. In EUROC dataset [4] we follow the same pre-processing step as [13] and use nearest interpolation to get dense depth map. In order to evaluate the trajectory quality we report the ATE-RMSE [cm] of all sequences. More details regarding convergence rate and run-time can be found in the supplementary material.

Method	Rm 0	Rm 1	Rm 2	Off 0	Off 1	Off 2	Off 3	Off 4	Avg
Vox-Fusion* [60]	1.37 0	4.70	1.47	8.48	2.04	2.58	1.11	2.94	3.09
ESLAM[20]	0.71 0	0.70	**0.52**	0.57	0.55	0.58	0.72	0.63	0.63
NICE-SLAM[63]	0.97 0	1.31	1.07	0.88	1.00	1.06	1.10	1.13	1.06
Ours	**0.53**	**0.45**	0.84	0.54	0.33	0.48	0.66	0.51	0.54
Ours(world)	0.62	0.52	0.91	0.60	0.62	0.36	0.54	0.72	0.58
Ours(rand)	35.84	9.29	34.67	N/A	9.69	26.92	N/A	N/A	N/A
Ours (intrinsic)	**0.53**	0.47	0.81	**0.35**	**0.24**	**0.43**	**0.64**	**0.50**	**0.49**

Table 5. **Tracking performance on Replica [51].** By integrating our method into the tracking branch of NICE-SLAM, we observe significant improvements. We investigate the impact of varying reference coordinates on tracking. It is evident that our proposed low DOF motion further improves the tracking performance.

Method	0000	0059	0106	0181	0207	Avg
DI-Fusion [16]	62.99	128.00	18.50	87.88	100.19	78.89
Vox-Fusion* [60]	68.84	24.18	8.41	23.30	9.41	26.90
NICE-SLAM[63]	12.00	14.00	7.90	13.40	6.20	10.70
Ours	**10.98**	11.98	**7.10**	13.50	5.76	9.86
Ours(intrinsic)	11.21	**8.78**	7.57	**12.21**	**4.87**	**8.93**

Table 6. **Tracking performance on ScanNet [9].** Our approach yields consistently better results than the baseline. Note that the gain of utilizing intrinsic motion is relatively small, possibly attributed to the challenges posed by the noisy ground truth poses.

Results. We report all tracking results using ATE-RMSE [cm]. The numbers for the baselines are taken from [47] except EUROC. We showcase the effectiveness of our method for tracking across all scenes in Table 5, 6, 7. We observe significant improvements in both relatively easy and challenging scenarios.

Moreover as illustrated in Table 5, we underscore the importance of defining the coordinate system for relative pose optimization. The tracking is unstable and difficult when fixed on world origin or random coordinates. Figure 6 further demonstrates that, through our estimation of intrinsic motion and its transformation with PoseNet, we attain pose within a low-dimensional manifold, resulting in a substantial enhancement of tracking performance.

Finally, we validate the effectiveness of our IMU-Fusion method. While baseline methods fail in the face of large illumination changes and noisy depth, our approach maintains robust tracking and achieves accuracy comparable to state-of-the-art sparse feature-based tracking methods.

Method	fr1/desk	fr2/xyz	fr3/office	Avg
DI-Fusion [16]	4.4	2.0	5.8	4.1
Vox-Fusion* [60]	3.52	1.49	26.01	10.34
NICE-SLAM[63]	4.26	31.73	3.87	13.28
Ours	2.97	7.38	3.76	4.70
Ours(intrinsic)	**2.72**	**1.98**	**2.74**	**2.48**

Table 7. **Tracking performance on TUM-RGBD [52]** Our method consistently outperforms NICE-SLAM and other dense neural RGBD methods. The effectiveness of intrinsic motion is also demonstrated for reducing the tracking error significantly.

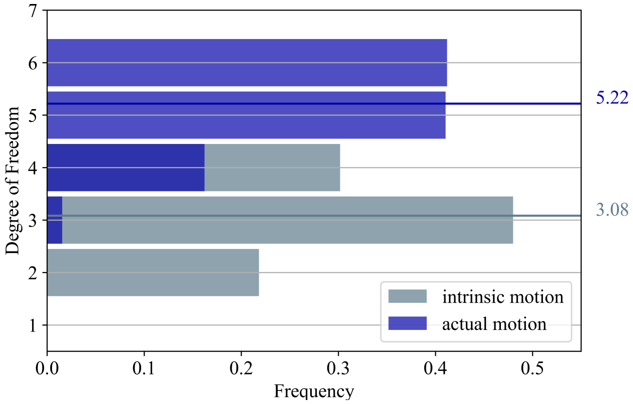


Figure 6. DOF Comparison on Replica room 1 dataset, we report that DOF of actual motion drop 41% from 5.22 to 3.08, demonstrating the sparsity of intrinsic motion.

Method	v101	v102	v103	v201	v202	v203	Avg
VINS-MONO[41]	7.9	11.0	18.0	8.0	16.0	27.0	14.6
ORB-SLAM [36]	**1.5**	2.0	N/A	2.1	1.8	N/A	N/A
DROID-SLAM [54]	3.7	**1.2**	**2.0**	**1.7**	**1.3**	**1.4**	**2.2**
NICE-SLAM[63]	2.58	N/A	5.66	6.56	N/A	N/A	N/A
Ours(loose)	2.20	6.74	**5.04**	**4.52**	3.87	19.06	6.77
Ours(tight)	1.98	**6.09**	5.55	4.99	**3.03**	**15.34**	**6.16**

Table 8. **Tracking performance on EUROC [4].** Our IMU-fusion improves tracking with lower error and robustness, outperforming NICE-SLAM. We report results with sparse tracking method for reference. Despite the gap, our method narrows differences with state-of-the-art sparse tracking.

5. Conclusion

We proposed a simple yet effective technique for optimizing camera pose as a continuous function of time. The benefits of this approach were illustrated through several experiments of diverse applications, namely NeRF from the inaccurate pose, NeRF using Event Cameras, and visual SLAM with Depth and IMUs. We also studied different designs of the time-to-pose mapping continuous function, leading us to prefer a decoupled architecture. Furthermore, we justified the ease of using the proposed PoseNet in a plug-and-play manner. We first propose IMU-Fusion in NeRF-SLAM and analyze the advantage of adopting intrinsic motion frame for camera tracking tasks. Clear advantages in terms of performance were also observed in all settings, thanks to the continuous motion prior.

Acknowledgements: Research is partially funded by VIVO Collaboration Project and the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

References

[1] Dejan Azinovic, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6280–6291. IEEE, 2022. 3

[2] Tim D Barfoot, Chi Hay Tong, and Simo Särkkä. Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Robotics: Science and Systems*, pages 1–10. Citeseer, 2014. 1, 3, 5

[3] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 3

[4] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35 (10):1157–1163, 2016. 7, 8

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 3

[6] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24–34, 2023. 3

[7] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8264–8273, 2023. 3, 5

[8] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. GARF: gaussian activated radiance fields for high fidelity reconstruction and pose estimation. *CoRR*, abs/2204.05735, 2022. 3

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7, 8

[10] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 3

[11] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016. 3

[12] Paul Furgale, Timothy D Barfoot, and Gabe Sibley. Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095. IEEE, 2012. 1, 3

[13] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019. 7

[14] Sachini Herath, David Caruso, Chen Liu, Yufan Chen, and Yasutaka Furukawa. Neural inertial localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6604–6613, 2022. 3

[15] 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. 7

[16] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8932–8941, 2021. 8

[17] Weibo Huang and Hong Liu. Online initialization and automatic camera-imu extrinsic calibration for monocular visual-inertial slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5182–5189. IEEE, 2018. 3

[18] 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. 3

[19] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Animashree Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields, 2021. 3

[20] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023. 8

[21] Lukas Kaul, Robert Zlot, and Michael Bosse. Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner. *Journal of Field Robotics*, 33(1):103–132, 2016. 3

[22] 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. 3

[23] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86. IEEE, 2009. 3

[24] 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. 3

[25] Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013. 3

[26] Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. *arXiv preprint arXiv:2301.08930*, 2023. 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. 2, 3, 4, 5, 6

[28] Qi Ma, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Deformable neural radiance fields using rgb and event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3590–3600, 2023. 3

[29] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6331–6341. IEEE, 2021. 3

[30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 1

[31] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 6

[32] 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, 3, 4, 6

[33] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. Towards visual slam with event-based cameras. In *The problem of mobile sensors workshop in conjunction with RSS*, 2015. 3

[34] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3565–3572. IEEE, 2007. 3

[35] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6): 1425–1440, 2018. 1

[36] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 8

[37] Janne Mustaniemi, Juho Kannala, Simo Särkkä, Jiri Matas, and Janne Heikkila. Gyroscope-aided motion deblurring with deep networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1914–1922. IEEE, 2019. 3

[38] Andreas Nüchter, Michael Bleier, Johannes Schauer, and Peter Janotta. Improving google's cartographer 3d mapping by continuous-time slam. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:543–549, 2017. 3

[39] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[40] Alonso Patron-Perez, Steven Lovegrove, and Gabe Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *International Journal of Computer Vision*, 113(3):208–219, 2015. 3

[41] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. 3, 8

[42] Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Emvs: Event-based multi-view stereo. 2016. 7

[43] A Richard, RA Newcombe, L Steven, et al. Dense tracking and mapping in real-time. In *Proceedings of IEEE International Conference on Computer Vision, Barcelona*, page 2327, 2011. 3

[44] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. *arXiv preprint arXiv:2210.13641*, 2022. 3

[45] 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. 7

[46] 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. 3, 4, 7

[47] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. *arXiv preprint arXiv:2304.04278*, 2023. 3, 8

[48] Davide Scaramuzza and Zichao Zhang. Visual-inertial odometry of aerial robots. *arXiv preprint arXiv:1906.03289*, 2019. 3

[49] Zhenmei Shi, Fuhao Shi, Wei-Sheng Lai, Chia-Kai Liang, and Yingyu Liang. Deep online fused video stabilization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1250–1258, 2022. 3

[50] Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017. 4

[51] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7, 8

[52] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012. 7, 8

[53] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6209–6218. IEEE, 2021. 3

[54] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 8

[55] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. *arXiv preprint arXiv:2211.11738*, 2022. 3

[56] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jörg Stückler, and Daniel Cremers. Visual-inertial mapping with non-linear factor recovery. *IEEE Robotics and Automation Letters*, 5(2):422–429, 2019. 3

[57] 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. 6

[58] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. 3

[59] Yitong Xia, Hao Tang, Radu Timofte, and Luc Van Gool. Sinerf: Sinusoidal neural radiance fields for joint pose estimation and scene reconstruction. *CoRR*, abs/2210.04553, 2022. 3

[60] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 8

[61] 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. 6

[62] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics*, 37 (5):1433–1450, 2021. 3

[63] 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. 2, 3, 4, 7, 8

[64] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint arXiv:2302.03594*, 2023. 3