

NeRF-VO: Real-Time Sparse Visual Odometry with Neural Radiance Fields

Jens Naumann¹ Binbin Xu² Stefan Leutenegger¹ Xingxing Zuo^{1,*}

jens.naumann@tum.de binbin.xu@utoronto.ca stefan.leutenegger@tum.de xingxing.zuo@tum.de

¹Technical University of Munich ²University of Toronto

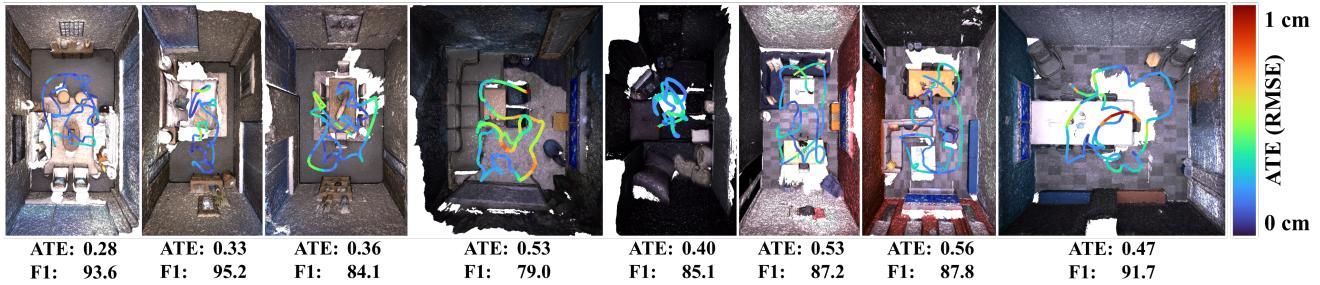


Figure 1. **3D reconstruction and camera tracking results on Replica** [36, 38]. Meshes rendered from optimized neural radiance fields. Scenes from left to right: room0-2, and office0-4. Quantitative evaluations of pose estimation and 3D reconstruction are reported in cm (ATE RMSE) and % (F1 score). NeRF-VO achieves an average ATE of 0.43 cm and F1 of 88.0 using solely RGB images as input.

Abstract

We introduce a novel monocular visual odometry (VO) system, NeRF-VO, that integrates learning-based sparse visual odometry for low-latency camera tracking and a neural radiance scene representation for sophisticated dense reconstruction and novel view synthesis. Our system initializes camera poses using sparse visual odometry and obtains view-dependent dense geometry priors from a monocular depth prediction network. We harmonize the scale of poses and dense geometry, treating them as supervisory cues to train a neural implicit scene representation. NeRF-VO demonstrates exceptional performance in both photometric and geometric fidelity of the scene representation by jointly optimizing a sliding window of keyframed poses and the underlying dense geometry, which is accomplished through training the radiance field with volume rendering. We surpass state-of-the-art methods in pose estimation accuracy, novel view synthesis fidelity, and dense reconstruction quality across a variety of synthetic and real-world datasets, while achieving a higher camera tracking frequency and consuming less GPU memory.

1. Introduction

Accurate pose estimation and 3D scene reconstruction of the environment using images are fundamental challenges in 3D computer vision and essential prerequisites for diverse applications in robotics and mixed reality. Neural Radiance Fields (NeRF) [25] have proven to be an excellent scene representation method for novel view synthesis tasks. The original NeRF employs large multi-layer perceptrons (MLPs) to decode 3D coordinates and ray directions into volume density and color, respectively. However, it fails to represent complex and large scenes, particularly those with fine details. Both training and rendering NeRFs with deep MLPs are computationally intensive and time-consuming. Recent works [4, 16, 26] have sought to accelerate NeRF by replacing deep MLPs with more efficient neural representations. For instance, Instant-NGP [26] achieves a substantial speedup by utilizing a hybrid representation that combines trainable multi-resolution hash encodings (MHE) with shared shallow MLPs. These accelerated neural radiance scene representations open the door for real-time critical SLAM/VO techniques.

NeRF-based scene representations enable high-fidelity photometric and geometric reconstruction and provide reasonable estimates for some unobserved regions, while being highly memory efficient. These representations enable the comprehensive utilization of information from raw im-

*Corresponding author: Xingxing Zuo.

ages. With neural volume rendering, every pixel is leveraged in the scene optimization process. This has potential to push the performance frontier beyond traditional direct [13–15, 59] and indirect SLAM [3, 6, 10, 18] methods. Lately, numerous works have aimed at integrating SLAM with neural implicit mapping [24, 34, 38, 45, 51, 57]. However, only a few focus on monocular RGB input [5, 21, 32, 54, 55, 58]. In terms of accuracy, RGB-only methods lag behind their RGB-D counterparts, especially in dense reconstruction. Furthermore, most of these approaches, regardless of input modality, are computationally expensive, lack real-time capability, and require significant GPU memory.

To address these issues, we propose NeRF-VO, a real-time capable sparse visual odometry with neural implicit dense mapping. A preview of its performance is provided in Fig. 1. We obtain the initial pose estimation and 3D sparse landmarks using low-latency learning-based sparse visual odometry. Up to scale dense geometric cues, including monocular dense depth and normals, are inferred using a transformer-based neural network. With the initial poses, camera-captured monocular images, and rough dense geometric priors, we can efficiently optimize a neural radiance field that implicitly represents the 3D scene. Accurate poses and dense geometry of the scene are recovered by minimizing the disparity of captured images and predicted dense geometric cues relative to the renderings generated from the neural radiance field. Hence, our proposed NeRF-VO comprises three main components: a sparse visual tracking front-end, a dense geometry enhancement module, and a NeRF-based dense mapping back-end. The system architecture is depicted in Fig. 2.

Overall, we introduce NeRF-VO, a neural SLAM system that employs sparse visual odometry for efficient pose estimation, and a NeRF scene representation for highly accurate dense mapping. It showcases superior geometric and photometric reconstruction in comparison to SOTA methods while maintaining the lowest tracking latency and GPU memory consumption among competing works. The main contributions of our work can be summarized as follows:

- We propose NeRF-VO, a monocular RGB SLAM system that utilizes sparse visual odometry for pose tracking and an implicit neural representation for mapping. This enables highly accurate camera tracking, promising 3D reconstruction, and high-fidelity novel view synthesis.
- We present a novel paradigm for optimizing a NeRF scene representation and camera poses by incorporating dense depth and surface normal supervision. Utilizing a transformer-based monocular depth network, we predict dense depth and surface normal priors. To align their scale with sparse visual odometry, we propose a dedicated sparse-to-dense scale alignment procedure.
- NeRF-VO demonstrates SOTA performance in camera tracking, 3D dense reconstruction, and novel view syn-

thesis across various synthetic and real-world datasets.

- We open-source our versatile neural SLAM framework, fostering further research by allowing easy integration of other neural representations and VO/SLAM methods.

In the remainder of this paper, we review related work in Sec. 2 and introduce the sparse visual tracking front-end in Sec. 3. We present the main methodology, which includes dense geometry enhancement, scale alignment, neural implicit mapping and dedicated NeRF optimization, in Sec. 4. We report and analyze extensive experimental results in Sec. 5. Finally, we conclude the paper and provide an overview of future work in Sec. 6.

2. Related Work

Learning-Based Visual Tracking. Our work centers on monocular visual odometry using RGB image sequences from a calibrated camera. This technique estimates the camera positions and orientations of each incoming frame. Unlike SLAM, visual odometry focuses on local coherence among consecutive frames and does not include SLAM’s loop closure optimization or global bundle adjustment. Recently, VO methods have advanced from traditional hand-crafted feature detection and matching modules to deep learning-based approaches, enhancing accuracy and robustness [29, 40–42, 46, 50]. Among all of them, DROID-SLAM [41] and DPVO [42] are two noteworthy works that leverage neural networks to predict the optical flow between consecutive images and iteratively update camera poses. DPVO [42] serves as the foundation for the front-end of our SLAM system since it’s highly efficient and accurate.

Dense Visual SLAM. Dense visual SLAM aims to construct a dense 3D representation of the environment instead of sparse 3D landmarks. Following the first real-time dense visual work DTAM [28], many approaches have been proposed, primarily those exploiting monocular depth prediction [7, 44, 56]. The scene representations selected in these works have also progressed from volumetric representations [28] to low-dimension latent representations [2, 7, 60] and the integration of pre-trained depth estimators [19, 40, 49]. In this work, we choose a volumetric neural radiance field as the representation due to its superior photometric and geometric accuracy.

NeRF-Enabled SLAM. Recently, many works have been proposed to integrate NeRF-based [25] neural implicit representations into SLAM. In general, existing methods can be differentiated into *one-stage* and *two-stage* approaches. *Two-stage* approaches use an existing SLAM algorithm as a tracking module to estimate depth maps and camera poses, and then use these estimates as supervisory signals to optimize an implicit neural representation as part of a mapping module. Early approaches such as Orbeez-SLAM [5] and NeRF-SLAM [32] demonstrated the ef-

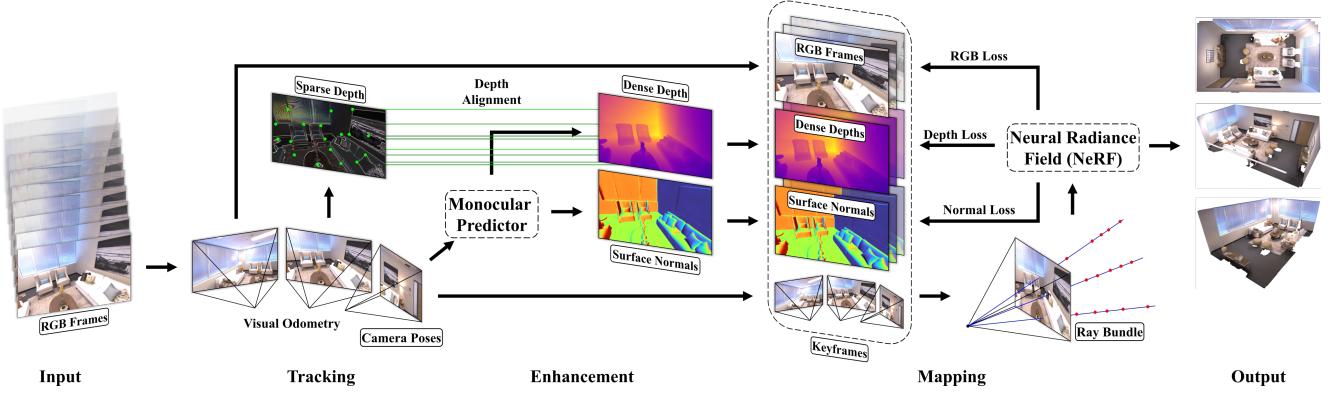


Figure 2. System architecture of NeRF-VO. The method uses only a sequence of RGB images as input. The sparse visual tracking module selects keyframes from this input stream and calculates camera poses and depth values for a set of sparse patches. Additionally, the dense geometry enhancement module predicts dense depth maps and surface normals and aligns them with the sparse depth from the tracking module. The NeRF-based dense mapping module utilizes raw RGB images, inferred depth maps, surface normals, and camera poses to optimize a neural implicit representation and refine the camera poses. Our system is capable of performing high-quality 3D dense reconstruction and rendering images at novel views.

fectiveness of this combination. Recent advancements in this direction have further introduced view-centric implicit functions [23], global loop closure [54, 55], and monocular depth priors [54] to improve pose estimation accuracy and dense mapping quality. In contrast, *one-step* approaches use a single implicit neural function for both tracking and mapping. iMAP [38] and NICE-SLAM [57] set the groundwork for this direction using RGB-D data. Subsequent works aimed at enhancing the scene representation [34, 45, 51], introducing implicit semantic encoding [24], and integrating inertial measurements [22]. Among these works, NICER-SLAM [58] and DIM-SLAM [21] perform dense SLAM only using monocular images. NICER-SLAM prioritizes high-fidelity scene reconstruction and novel view synthesis through heavy optimization with various losses but is unsuitable for real-time applications. DIM-SLAM, on the other hand, focuses on accurate camera tracking, yet shows suboptimal performance in dense reconstruction.

Our work follows a two-stage design. We use a sparse visual tracking method, DPVO [42] to get initial poses at high frequency and obtain dense depth and surface normals priors from a mono-depth prediction network. These initial poses and dense geometry cues are used to train our mapping back-end employing a nerfacto-based neural radiance field [39]. Our system yields high-frequency pose estimations and promising dense reconstruction after neural scene optimization, all achieved at a low memory footprint.

3. Sparse Visual Pose Tracking

We employ the Deep Patch Visual Odometry (DPVO) [42] algorithm as our tracking front-end. DPVO is a sparse, monocular, learning-based algorithm that estimates camera poses and sparse depths for a set of patches per keyframe.

Patch Graph. Given a sequence of RGB frames, DPVO randomly samples a set of K square patches of size s per keyframe and adds them to a bipartite patch graph that connects patches and frames. For instance, the k -th square patch from frame i represented by $\mathbf{P}_k^i = [\mathbf{u} \ \mathbf{v} \ \mathbf{1} \ \mathbf{d}]^T$ is connected via edges to all frames within temporal vicinity of frame i . \mathbf{u}, \mathbf{v} represent the pixel coordinates and \mathbf{d} denotes the inverse depths with $\mathbf{u}, \mathbf{v}, \mathbf{d} \in \mathbb{R}^{s^2}$. $\mathbf{1}$ is a vector filled with ones. Thus, the patch graph builds a trajectory for each patch, incorporating all of its reprojections \mathbf{P}_k^{ij} , where $j \in \mathcal{N}$, and \mathcal{N} is the array of frames temporally adjacent to i . Assuming uniform depth across each patch, the reprojection is defined by:

$$\mathbf{P}_k^{ij} \sim \mathbf{K} \mathbf{T}_j \mathbf{T}_i^{-1} \mathbf{K}^{-1} \mathbf{P}_k^i, \quad (1)$$

where \mathbf{K} is the camera calibration matrix, and $\mathbf{T}_i \in \text{SE}(3)$ represents the world-to-camera transformation of frame i . For simplicity, we omit the normalization operation of the third dimension in the above equation.

Differentiable Bundle Adjustment. The key component of DPVO is its differentiable pose and depth update operator. A recurrent neural network operates on the patch graph with a set of edges \mathcal{E} , while maintaining a hidden state for each edge $(k, i, j) \in \mathcal{E}$ (patch-frame pair), and predicting a 2D correction vector $\delta_k^{ij} \in \mathbb{R}^2$ for each reprojection of the patch center with a corresponding confidence weight $\psi_k^{ij} \in \mathbb{R}^2$. Bundle adjustment is performed using the optical flow correction as a constraint to iteratively update frame poses and patch depths via nonlinear least-squares optimization. The cost function for bundle adjustment is:

$$\sum_{(k,i,j) \in \mathcal{E}} \left\| \mathbf{K} \mathbf{T}_j \mathbf{T}_i^{-1} \mathbf{K}^{-1} \bar{\mathbf{P}}_{ki} - \left[\bar{\mathbf{P}}_k^{ij} + \delta_k^{ij} \right] \right\|_{\psi_k^{ij}}^2, \quad (2)$$

where $\|\cdot\|_\psi$ denotes the Mahalanobis distance and $\bar{\mathbf{P}}$ denotes the patch center. The bundle adjustment step of DPVO is differentiable, the recurrent neural network is trained on a set of ground truth poses, together with the corresponding ground truth optical flow to supervise pose estimation and optical flow correction.

Keyframing Strategy. DPVO optimizes the patch depths and camera poses of all keyframes in a sliding optimization window, adding each incoming frame. The sliding window keeps a constant array of the most recent keyframes and removes old ones to maintain bounded computational complexity and to ensure runtime efficiency. The three most recent frames are always considered to be keyframes, but the fourth most recent keyframe is only kept if the optical flow to its predecessor is sufficiently high, otherwise, it is removed. We use this keyframe selection for our downstream dense geometry enhancement and mapping modules. To stabilize NeRF optimization, the three most recent keyframes are not considered, as they are likely to be removed later by the keyframing scheme described above. Each time the fourth most recent keyframe is secured, it is propagated to our dense geometry enhancement module and NeRF optimization, including all previously active camera poses and patches in the current sliding window. Since only certain keyframes are involved in our dense geometry enhancement and NeRF optimization, the computational and memory footprint of NeRF-VO are bounded.

4. Neural Implicit Dense Mapping

As indicated in the system overview in Fig. 2, the keyframes from the aforementioned visual tracking front-end are subsequently processed by our dense geometry enhancement and incorporated into our NeRF optimization back-end.

4.1. Dense Geometry Enhancement

Since the sparse patch depths \mathbf{D}_s from the visual tracking module cover only a tiny portion of the image, we incorporate a dense geometry enhancement module that predicts a dense depth map \mathbf{D}_d based solely on the monocular RGB input. We then align this dense depth prediction to the sparse patch depths using a dedicated scale alignment procedure. Monocular depth prediction is performed using the off-the-shelf network [12] based on the Dense Prediction Transformer (DPT) architecture [30, 31].

Scale Alignment. Since both sparse and dense depths are predicted solely based on RGB input, their depth distributions are skewed relative to the metric depth and to each other. However, downstream dense mapping necessitates depth maps and camera poses at a unified scale. Therefore, we introduce a dense-to-sparse scale alignment procedure to align the predicted dense depth map of each keyframe to the sparse patch depth. Assuming that the sparse depth \mathbf{D}_s

and the dense depth \mathbf{D}_d follow a Gaussian distribution with $\mathbf{D}_s \sim \mathcal{N}(\mu_s, \sigma_s^2)$ and $\mathbf{D}_d \sim \mathcal{N}(\mu_d, \sigma_d^2)$, we compute the aligned dense depth by:

$$\mathbf{D}'_d = \alpha \mathbf{D}_d + \beta \mathbf{1}, \quad (3)$$

where α and β are a scale and shift value defined as:

$$\alpha = \frac{\sigma_s}{\hat{\sigma}_d}, \quad \beta = \mu_d \left(\frac{\mu_s}{\hat{\mu}_d} - \alpha \right), \quad (4)$$

where $\hat{\mu}_d$ and $\hat{\sigma}_d$ are the mean and standard deviation statistics of the sparsified dense depth map $\hat{\mathbf{D}}_d$ extracted from \mathbf{D}_d at the pixel coordinates of \mathbf{D}_s . Since the pixel coordinates are sampled randomly, given a sufficiently large sample size per frame, one could assume that $\mu_d \approx \hat{\mu}_d$. This would lead to a relaxed formulation of β with $\beta = \mu_s - \alpha \hat{\mu}_d$. We have empirically evaluated this relaxed alignment scheme along with several other alignment strategies and found that the standard formulation in Eq. 4 performs best (see Sec. 5.8).

Outlier Rejection. Due to the randomness in the patch selection, some of the patches are sampled on textureless or non-Lambertian surfaces, causing the bundle adjustment to fail or to produce poor depth estimates. Thus, we introduce an outlier rejection scheme where the patches with depth values in the upper and lower 1/12th percentile per frame are excluded from the scale alignment.

Surface Normal Cues. To improve the accuracy of the neural implicit representation for fine-grained dense mapping, we introduce surface normal constraints by matching against surface normals predicted from the RGB input using the same off-the-shelf monocular predictor from [12].

4.2. Mapping with Neural Scene Representation

Neural Radiance Field. Our dense mapping back-end reconstructs the scene by optimizing a neural radiance field built on the nerfacto model [39]. The neural scene representation comprises multi-resolution hash encodings similar to [26], spherical harmonics encoding, and two multi-layer perceptrons (MLPs) for density and color recovery. The NeRF scene representation is optimized using the camera poses, captured RGB images as well as inferred depth maps and surface normals of all keyframes. How keyframes are added to this optimization is explained in Sec. 3.

We generate ray bundles \mathcal{R} inside the camera frustums of the keyframes and sample 3D coordinates along these rays $t \in \mathcal{R}$ by using a proposal sampler, which is optimized jointly with the rest of the architecture. Each sampled 3D coordinate is mapped to a set of learnable embeddings in the multi-resolution hashmap. These embeddings are processed by the first MLP to predict the volume density. The resulting density is fed into the second MLP together with the spherical harmonic encoded view direction and appearance embeddings to compute the color. The surface normals

per sample are obtained by computing the gradient over the first MLP with respect to the hashed 3D coordinate. Finally, the color, depth, and surface normal for the entire ray are obtained using volumetric rendering similar to [25].

Joint Optimization of Poses and Neural Scene. Since the keyframe camera poses \mathbf{T}_C are involved in ray sampling, they can be optimized jointly with the NeRF scene representation. Hence, we optimize the learnable parameters Θ of the scene representation and the MLPs jointly with the camera poses \mathbf{T}_C using a weighted sum of four losses.

Our color loss is defined as the mean squared error between the rendered color image and the ground truth RGB input, analogous to [25]:

$$\mathcal{L}_{\text{rgb}}(\mathbf{T}_C, \Theta) = \sum_{(u,v) \in \mathcal{B}} \left\| \mathbf{C}_{uv} - \check{\mathbf{C}}_{uv}(\mathbf{T}_C, \Theta) \right\|_2^2, \quad (5)$$

where $\check{\mathbf{C}}_{uv}$ and \mathbf{C}_{uv} are the rendered and ground truth color, and (u, v) are the pixel coordinates of the ray bundle \mathcal{R} .

To counter noise and bias in our depth estimation and alignment procedures, we employ the uncertainty-aware depth loss [11], which aims to minimize the Kullback-Leibler divergence between the assumed Gaussian distributed depth values of all samples along each ray and our aligned dense depth map \mathbf{D}'_d :

$$\begin{aligned} \mathcal{L}_d(\mathbf{T}_C, \Theta) = & \sum_{(u,v) \in \mathcal{B}} \sum_{t \in \mathcal{R}_{uv}} \log(\mathcal{T}_t(\mathbf{T}_C, \Theta)) \cdot \\ & \exp\left(-\frac{(d_t - \mathbf{D}'_{d_{uv}})^2}{2\hat{\sigma}^2}\right) \Delta d_t, \end{aligned} \quad (6)$$

where $t \in \mathcal{R}_{uv}$ is the t -th sample along the ray \mathcal{R}_{uv} , d_t is the distance of the sample t from the camera center along ray \mathcal{R}_{uv} , Δd_t is the distance between the sampling distances d_{t+1} and d_t with $\Delta d_t = d_{t+1} - d_t$, $\hat{\sigma}$ is the estimated variance of the depth $\mathbf{D}'_{d_{uv}}$ (we use 0.001 empirically) and $\mathcal{T}_t(\mathbf{T}_C, \Theta)$ is the accumulated transmittance along the ray \mathcal{R}_{uv} up to sample t computed by:

$$\mathcal{T}_t(\mathbf{T}_C, \Theta) = \exp\left(-\sum_{s < t} \rho_s(\mathbf{T}_C, \Theta) \Delta d_s\right), \quad (7)$$

where $\rho_s(\cdot)$ is the estimated density at the sample s .

The normal supervision is guided by the normal consistency loss proposed in [52], which integrates the L1 distance and cosine distance between the rendered surface normals $\check{\mathbf{N}}_{uv} \in \mathbb{R}^3$ and the normals predicted from the monodepth network \mathbf{N}_{uv} at pixel (u, v) :

$$\begin{aligned} \mathcal{L}_n(\mathbf{T}_C, \Theta) = & \sum_{(u,v) \in \mathcal{B}} \left\| \mathbf{N}_{uv} - \check{\mathbf{N}}_{uv}(\mathbf{T}_C, \Theta) \right\|_1 + \\ & \left\| 1 - \mathbf{N}_{uv}^\top \check{\mathbf{N}}_{uv}(\mathbf{T}_C, \Theta) \right\|_1, \end{aligned} \quad (8)$$

where $\mathbf{N}_{uv}^\top \check{\mathbf{N}}_{uv}$ indicates the cosine similarity.

Finally, we use two regularization terms for the volume density. The distortion and proposal losses introduced by [1] serve to prevent floaters and background collapse and guide the optimization of the proposal sampler:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{prop}} + 0.002\mathcal{L}_{\text{dist}}, \quad (9)$$

Overall, the complete loss for our NeRF optimization is:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + 0.001\mathcal{L}_d + 0.00001\mathcal{L}_n + \mathcal{L}_{\text{reg}} \quad (10)$$

4.3. System Design

The sparse tracking, dense geometry enhancement, and dense mapping modules run asynchronously on separate threads. Information is propagated in a feed-forward fashion through the three components sequentially. Communication occurs each time a new keyframe is secured in the visual tracking module. It is then processed by the enhancement module and included in the training database of the mapping module. Separated from this process, the mapping module continuously optimizes the implicit representation jointly with the camera poses. With this multi-threaded architecture, our system can run in real-time, with the visual tracking running at low latency, while the enhancement and mapping run at a relatively low frequency, which does not block the data streamed in from the sensor. Due to its asynchronous architecture, our system has the potential to be run on multiple GPUs, leading to improved runtime efficiency.

5. Experiments

We evaluate our method qualitatively and quantitatively against the state-of-the-art monocular SLAM methods using neural implicit representations on a variety of real and synthetic indoor datasets, and present an extensive ablation study of the key design choices of our method.

5.1. Evaluation Datasets

We evaluate our method on four datasets. Replica [36]: a synthetic dataset that contains high-quality reconstructions of indoor scenes, where we use the eight trajectories generated by [38]. ScanNet [8]: a real-world dataset where the RGB-D sequence is captured using an RGB-D sensor attached to a handheld device and the ground truth trajectory is computed using BundleFusion [9]. 7-Scenes [35]: a real-world dataset that provides a trajectory of RGB-D frames captured by Kinect sensors where the ground truth trajectories and dense 3D models are computed using KinectFusion [27]. We also evaluate our method on a self-captured dataset consisting of a set of four RGB-D trajectories captured with the RealSense sensor in a meeting room. The ground-truth trajectories are computed using the visual-inertial SLAM system OKVIS2 [20] based on synchronized stereo images and IMU data, while the 3D model is obtained using TSDF-Fusion [27] on the captured depth maps.

5.2. Baselines

We compare our method against the SOTA SLAM method NeRF-SLAM [32], which also uses neural implicit representations for mapping. Additionally, where applicable, we report results from the contemporary methods DIM-SLAM [21], NICER-SLAM [58], Orbeez-SLAM [5], GO-SLAM [55] and HI-SLAM [54]. To evaluate the benefits of our back-end, we compare the camera tracking accuracy against DPVO [42] as well as the visual odometry version of DROID-SLAM [41] (DROID-VO), which is used as the tracking module in NeRF-SLAM. These models are evaluated under the same conditions as ours.

5.3. Metrics

We evaluate our method on three tasks: camera tracking, novel view synthesis, and 3D dense reconstruction. For camera tracking, we align the estimated trajectory to the ground truth using the Kabsch-Umeyama [17, 43] algorithm and evaluate the accuracy using the absolute trajectory error (ATE RMSE) [37]. For novel view synthesis, we assess the quality of the rendered RGB images using peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [47], and learned perceptual image patch similarity (LPIPS) [53]. To compute these metrics, we select 125 equally spaced RGB frames along the trajectory. To render the exact view of these evaluation frames, we transform the ground truth camera poses to the coordinate system of the implicit function. For 3D reconstruction, we consider accuracy, completion, and recall. We render the predicted mesh from our neural implicit representation. Analogous to [21, 32, 54, 55, 58] we exclude regions not observed by any camera from the evaluation. We align both meshes using the Iterative Closest Point (ICP) [33] algorithm. To ensure statistical validity, we report the average of 5 independent runs.

5.4. Camera Tracking

Table 1 reports the camera tracking results on the Replica dataset [36, 38], showing that our model achieves the lowest average ATE RMSE over all scenes. The results of DIM-SLAM [21] and NICER-SLAM [58] are taken from their original papers. Other methods have been evaluated by us under exactly the same conditions as ours. While DIM-SLAM [21] and NeRF-SLAM [32] perform better on some scenes, their performance deteriorates on others, whereas our method remains relatively stable across all scenes.

Table 2 reports the tracking accuracy on six commonly tested scenes from ScanNet [8]. Our method outperforms concurrent visual odometry methods whose tracking module only applies implicit “local loop closure” (L-LC) imposed by the scene representations, such as HI-SLAM (VO), DROID-VO, NeRF-SLAM, and DPVO. In general, we can conclude that the full SLAM methods incorporating

Table 1. Camera tracking performance (ATE RMSE [cm]) on the Replica [36] dataset.

Model	rm-0	rm-1	rm-2	of-0	of-1	o-2	of-3	of-4	Avg.
DIM-SLAM [21]	0.48	0.78	0.35	0.67	0.37	0.36	0.33	0.36	0.46
NICER-SLAM [58]	1.36	1.60	1.14	2.12	3.23	2.12	1.42	2.01	1.88
DROID-VO [41]	0.50	0.70	0.30	0.98	0.29	0.84	0.45	1.53	0.70
NeRF-SLAM [32]	0.40	0.61	0.20	0.21	0.45	0.59	0.33	1.30	0.51
DPVO [42]	0.49	0.54	0.54	0.77	0.36	0.57	0.46	0.57	0.54
Ours	0.28	0.33	0.36	0.53	0.40	0.53	0.56	0.47	0.43

Table 2. Camera tracking performance (ATE RMSE [cm]) on ScanNet [8]. The methods are categorized according to whether they perform global (G) or only local (L) loop closure (LC) optimization.

Model		0000	0059	0106	0169	0181	0207	Avg.
G-LC	Orbeez-SLAM [5]	7.2	7.2	8.1	6.6	15.8	7.2	8.7
	GO-SLAM [55]	5.9	8.3	8.1	8.4	8.3	n/a	n/a
	HI-SLAM [54]	6.4	7.2	6.5	8.5	7.6	8.4	7.4
L-LC	HI-SLAM (VO) [54]	14.4	26.7	10.0	15.5	9.3	9.9	14.3
L-LC	DROID-VO [41]	14.4	16.4	10.9	16.3	10.8	13.6	13.7
	NeRF-SLAM [32]	14.9	16.6	10.7	16.5	12.8	13.8	14.2
	DPVO [42]	13.1	19.2	12.4	13.3	8.8	8.9	12.6
	Ours	12.7	19.0	12.4	13.2	9.0	8.6	12.5

Table 3. Camera tracking performance (ATE RMSE [cm]) on our custom dataset.

Model	seq-1	seq-2	seq-3	seq-4	Avg.
DROID-VO [41]	2.58	1.37	1.81	3.05	2.20
NeRF-SLAM [32]	2.32	1.19	1.31	5.26	2.52
DPVO [42]	2.96	1.38	1.57	3.14	2.26
Ours	2.08	1.07	1.43	2.68	1.81

global loop closure (G-LC) and scene-wide global bundle adjustment perform better on these large scenes.

Finally, Table 3 shows the performance of our method and NeRF-SLAM [32] as well as their respective tracking modules on our custom dataset, demonstrating superior camera tracking performance of our model.

5.5. 3D Reconstruction

Table 4 demonstrates the superior 3D reconstruction performance of our model over all compared concurrent works on the Replica [36] dataset. Our method outperforms SOTA in accuracy, completion, and recall on four scenes and on average. The inferior performance on the office0 and office1 scenes is the result of scene-specific characteristics that complicate monocular depth predictions: these scenes have detail-rich wall paintings that lead to artifacts in monocular depth estimation. We notice that this is also the case for NICER-SLAM [58], which similarly relies on a monocular depth prediction network. A qualitative comparison of the 3D reconstruction is provided in Fig. 3.

Table 5 reports the 3D reconstruction performance of our method and NeRF-SLAM [32] on three real-world datasets: 7-Scenes [35], ScanNet [8], and our custom dataset. The



Figure 3. 3D reconstructions of five scenes from Replica [36]. The pictures of GO-SLAM [55] and HI-SLAM [54] have been taken from their respective papers.

results show that our model outperforms NeRF-SLAM on almost all scenes and on average on all datasets. The scene-level results are available in the supplementary material. Unfortunately, none of the concurrent papers report 3D reconstruction performance on any dataset other than the synthetic Replica [36], thus we are unable to benchmark our model against other methods besides NeRF-SLAM.

5.6. Novel View Synthesis

We evaluated our model for novel view synthesis on Replica [36]. The results can be found in table 6. They indicate mixed performance on the different metrics. While NICER-SLAM performs best on SSIM, NeRF-SLAM and our method are almost on par. NeRF-SLAM performs best in PSNR, yet our approach is almost equal in most scenes.

Table 4. 3D reconstruction results on the Replica [36] dataset.

Model	Metric	rm-0	rm-1	rm-2	of-0	of-1	of-2	of-3	of-4	Avg.
DIM-SLAM [21]	Acc. [cm] ↓	3.68	3.64	5.84	2.60	2.02	4.50	5.43	4.57	4.03
	Com. [cm] ↓	5.32	4.72	5.70	2.65	3.31	6.09	5.98	5.81	4.20
	Rec. [$<5\text{cm}$ %] ↑	82.2	80.4	74.44	89.6	84.9	75.3	73.5	76.6	79.6
NICER-SLAM [58]	Acc. [cm] ↓	2.53	3.93	3.40	5.49	3.45	4.02	3.34	3.03	3.65
	Com. [cm] ↓	3.04	4.10	3.42	6.09	4.42	4.29	4.03	3.87	4.16
	Rec. [$<5\text{cm}$ %] ↑	88.8	76.6	86.1	65.2	77.8	74.5	82.0	84.0	79.4
GO-SLAM [55]	Acc. [cm] ↓	4.60	3.31	3.97	3.05	2.74	4.61	4.32	3.91	3.81
	Com. [cm] ↓	5.56	3.48	6.90	3.31	3.46	5.16	5.40	5.01	4.79
	Rec. [$<5\text{cm}$ %] ↑	73.4	82.9	74.2	82.6	86.2	75.8	72.6	76.6	78.0
HI-SLAM [54]	Acc. [cm] ↓	3.33	3.50	3.11	3.77	2.46	4.86	3.92	3.53	3.56
	Com. [cm] ↓	3.29	3.20	3.39	3.65	3.61	3.68	4.13	3.82	3.60
	Rec. [$<5\text{cm}$ %] ↑	86.4	85.8	83.0	80.7	82.4	82.9	80.3	82.3	83.0
NeRF-SLAM [32]	Acc. [cm] ↓	2.77	4.50	3.45	1.88	2.09	3.77	3.24	3.06	3.10
	Com. [cm] ↓	3.45	3.49	6.32	3.76	3.19	4.20	4.23	3.97	4.08
	Rec. [$<5\text{cm}$ %] ↑	90.2	83.1	82.3	88.4	85.7	82.4	87.0	87.1	85.8
Ours	Acc. [cm] ↓	2.24	1.89	3.02	3.45	3.15	3.30	3.05	2.35	2.81
	Com. [cm] ↓	2.96	2.33	6.04	3.83	3.15	3.41	3.64	3.36	3.59
	Rec. [$<5\text{cm}$ %] ↑	91.6	93.0	81.5	79.2	83.5	86.4	86.1	88.7	86.3

Table 5. Averaged 3D reconstruction results on the real-world datasets: 7-Scenes [35], ScanNet [8], and our own dataset.

Model	Metric	7-Scenes	ScanNet	Custom
NeRF-SLAM [32]	Acc. [cm] ↓	20.91	19.2	13.09
	Com. [cm] ↓	26.81	25.6	10.16
	Rec. [$<5\text{cm}$ %] ↑	28.5	21.1	47.6
Ours	Acc. [cm] ↓	20.11	17.3	4.73
	Com. [cm] ↓	21.62	23.1	6.75
	Rec. [$<5\text{cm}$ %] ↑	34.0	22.5	65.0

Table 6. Novel view synthesis results on Replica [36, 38].

Model	Metric	rm-0	rm-1	rm-2	of-0	of-1	of-2	of-3	of-4	Avg.
NICER-SLAM [58]	PSNR ↑	25.33	23.92	26.12	28.54	25.86	21.95	26.13	25.47	25.41
	SSIM ↑	0.751	0.771	0.831	0.866	0.852	0.820	0.856	0.865	0.827
	LPIPS ↓	0.250	0.215	0.176	0.172	0.178	0.195	0.162	0.177	0.191
NeRF-SLAM [32]	PSNR ↑	34.07	34.12	37.06	40.36	39.27	36.45	36.73	37.69	36.97
	SSIM ↑	0.724	0.652	0.830	0.903	0.860	0.777	0.809	0.855	0.799
	LPIPS ↓	0.185	0.266	0.284	0.180	0.111	0.159	0.143	0.194	0.190
Ours	PSNR ↑	34.57	35.15	36.02	37.59	38.04	36.19	36.09	37.36	36.38
	SSIM ↑	0.768	0.760	0.779	0.874	0.823	0.776	0.791	0.834	0.801
	LPIPS ↓	0.159	0.114	0.119	0.051	0.044	0.106	0.097	0.107	0.100



Figure 4. Ground truth images (top) and images rendered by our model (bottom) of the room2 (left) and room0 (right) scenes from Replica [36]. Notably, there is minimal visual disparity between the rendered images and the original scene, even when viewed at close range.

Considering the LPIPS metric, our model shows a significantly superior performance on all scenes, indicating a high subjective similarity for a human observer, underlining the photometric strength of our approach. This can also be observed in the qualitative results in Fig. 4, where the rendered image and ground truth are almost indistinguishable.

Table 7. Runtime and maximum GPU memory utilization on the Replica [36, 38] dataset. Information on other methods is taken from the respective original papers. “ \leq ” and “ \simeq ” indicate missing details on maximum GPU memory usage. In such cases, we have reported the memory capacity of the device used.

Model	Tracking [fps] \uparrow	Mapping [fps] \uparrow	GPU Memory [gb] \downarrow
GO-SLAM [55]	8	8	18
DIM-SLAM [21]	14	3	\simeq 11
NICER-SLAM [58]	7	2	\leq 40
NeRF-SLAM [32]	15	10	\simeq 11
Ours	20	6	9

Table 8. Impact of using the aligned dense depths from the enhancement module vs. directly using the sparse depth from the tracking module. Results on the Replica [36, 38] dataset.

	ATE [cm] \downarrow	Acc. [cm] \downarrow	Com. [cm] \downarrow	Rec. [$<5\text{cm}$] \uparrow
sparse depth	0.48	8.16	6.49	72.5
dense depth	0.43	2.81	3.59	86.3

5.7. Runtime Analysis

We have conducted a runtime and memory utilization analysis of our method. A comparison with concurrent work is presented in Table 7. The results indicate that our model achieves the highest tracking FPS while leaving the smallest GPU memory footprint. This is primarily due to our runtime- and memory-efficient sparse visual tracking module, and dense geometry enhancement paradigm. In terms of mapping runtime, our method is in the midfield of all competing works. However, it must be emphasized that the two faster methods GO-SLAM [55] and NeRF-SLAM [32] rely on the highly optimized NeRF implementation by Müller *et al.* [26] using custom CUDA kernels, while our work, similar to DIM-SLAM [21] and NICER-SLAM [58], uses an unoptimized Python implementation, yet our model achieves twice the FPS compared to these Python-based alternatives. Finally, we have also conducted an ablation study on the impact of skipping input frames to reduce runtime, which demonstrates stable and promising performance even with a fourfold speedup (see Sec. 5.8).

5.8. Ablations

Dense Depths. We performed an ablation to validate that using aligned dense depths from a monocular estimator does indeed improve performance over directly using the sparse depth patches from our tracking module. The results can be found in Table 8. The substantial performance improvement from using dense depths indicates that optimizing a NeRF with only a few pixel depths per frame is not sufficient to achieve SOTA results.

Scale Alignment. As discussed in Sec. 4, the sparse depths from our tracking module and the dense depths from

Table 9. Impact of various scale alignment strategies on camera tracking and 3D reconstruction on the Replica [36, 38] dataset.

	ATE \downarrow	Acc. \downarrow	Com. \downarrow	Rec. \uparrow
none	0.69	9.13	7.06	66.0
min-max	0.43	8.72	9.59	65.1
least-squares [48]	0.45	3.86	4.41	77.3
ours (relaxed)	0.45	3.57	4.04	82.9
ours	0.43	2.81	3.59	86.3

Table 10. Impact of frame skipping on camera tracking and 3D reconstruction on the Replica [36, 38] dataset.

Total Frames	Speedup	ATE \downarrow	Acc. \downarrow	Com. \downarrow	Rec. \uparrow
100%	$\times 1$	0.43	2.81	3.59	86.3
50%	$\times 2$	0.46	2.85	3.61	86.6
25%	$\times 4$	0.50	2.83	3.70	85.9
12.5%	$\times 8$	3.88	6.34	6.13	77.1

the monocular estimator are skewed relative to each other, which requires an alignment strategy. In Table 9 we compare five linear alignment schemes: No alignment of sparse and dense depths (none), alignment based on the minimum and maximum of each depth map (min-max), the least squares alignment proposed by [48] (least-squares), our probability-based alignment method (ours) as well as its relaxed alternative (see Sec. 4). Our choice was confirmed as the probability-based alignment performs best.

Frame Skipping. Analogous to the authors of GO-SLAM [55], we performed a frame skipping ablation to assess what percentage of the total frames is necessary to still achieve SOTA performance. This has a direct impact on the runtime and performance of our model. The results are shown in Table 10. We discovered that, even with 25% of all frames, our approach still achieves nearly equivalent results, with performance degrading only after a reduction to 12.5% (every 8th frame). This means that skipping 75% of the frames could lead to a fourfold speedup of the model, allowing its use in real-time applications.

6. Conclusion

We introduce NeRF-VO, a novel neural visual odometry system that combines a learning-based sparse visual odometry for pose tracking, a monocular-depth prediction network for inferring dense geometry cues, and a specifically designed neural radiance field optimization for pose and dense geometry refinement. NeRF-VO surpasses SOTA methods, demonstrating superior pose estimation accuracy and delivering high-quality dense mapping, all while maintaining low pose tracking latency and GPU memory consumption. As part of future research, it would be compelling to integrate visual or geometric constraints obtained from the neural scene representation into the pose tracking front-end. This integration has the potential to further reduce drift and improve the accuracy of initial pose tracking.

References

- [1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 5
- [2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam — learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. 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. 2
- [4] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, page 333–350, Berlin, Heidelberg, 2022. Springer-Verlag. 1
- [5] Chi-Ming Chung, Yang-Che Tseng, Ya-Ching Hsu, Xiang-Qian Shi, Yun-Hung Hua, Jia-Fong Yeh, Wen-Chin Chen, Yi-Ting Chen, and Winston H. Hsu. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9400–9406, 2023. 2, 6
- [6] Igor Cvišić, Ivan Marković, and Ivan Petrović. Soft2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric. *IEEE Transactions on Robotics*, 39(1):273–288, 2023. 2
- [7] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J. Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020. 2
- [8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6, 7
- [9] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3), 2017. 5
- [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. 2
- [11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12882–12891, 2022. 5
- [12] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10786–10796, 2021. 4
- [13] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision – ECCV 2014*, pages 834–849, Cham, 2014. Springer International Publishing. 2
- [14] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018.
- [15] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. 2
- [16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinzhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022. 1
- [17] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 34(5):827–828, 1978. 6
- [18] Georg Klein and David William Murray. Parallel tracking and mapping for small AR workspaces. In *Sixth IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, 13–16 November 2007, Nara, Japan*, pages 225–234. IEEE Computer Society, 2007. 2
- [19] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. TANDEM: tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning, 8–11 November 2021, London, UK*, pages 34–45. PMLR, 2021. 2
- [20] Stefan Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure, 2022. 5
- [21] Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. In *Proceedings of the International Conference on Learning Representations*, 2023. 2, 3, 6, 7, 8
- [22] D. Lisus, C. Holmes, and S. Waslander. Towards open world nerf-based slam. In *2023 20th Conference on Robots and Vision (CRV)*, pages 37–44, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3
- [23] Hidenobu Matsuki, Keisuke Tateno, Michael Niemeyer, and Federico Tombari. Newton: Neural view-centric mapping for on-the-fly large-scale slam, 2023. 3
- [24] Kirill Mazur, Edgar Sucar, and Andrew J. Davison. Feature-realistic neural fusion for real-time, open set scene understanding. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8201–8207, 2023. 2, 3
- [25] 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. In *Computer Vision – ECCV 2020*, pages 405–421, Cham, 2020. Springer International Publishing. 1, 2, 5
- [26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. 1, 4, 8

- [27] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. 5
- [28] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 International Conference on Computer Vision*, pages 2320–2327, 2011. 2
- [29] Chethan M. Parameshwara, Gokul Hari, Cornelia Fermüller, Nitin J. Sanket, and Yiannis Aloimonos. Diffposenet: Direct differentiable camera pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6845–6854, 2022. 2
- [30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 4
- [31] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2022. 4
- [32] Antoni Rosinol, John J. Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields, 2022. 2, 6, 7, 8
- [33] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 6
- [34] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18433–18444, 2023. 2, 3
- [35] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene co-ordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 5, 6, 7
- [36] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces, 2019. 1, 5, 6, 7, 8
- [37] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. 6
- [38] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6229–6238, 2021. 1, 2, 3, 5, 6, 7, 8
- [39] Matthew Tancik, Ethan Weber, Evonne Ng, Rui long Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, New York, NY, USA, 2023. Association for Computing Machinery. 3, 4
- [40] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [41] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, pages 16558–16569. Curran Associates, Inc., 2021. 2, 6
- [42] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry, 2023. 2, 3, 6
- [43] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 6
- [44] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Niklaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [45] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Coslam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13293–13302, 2023. 2, 3
- [46] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tar-tanvo: A generalizable learning-based vo. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1761–1772. PMLR, 2021. 2
- [47] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6
- [48] Diana Wofk, René Ranftl, Matthias Müller, and Vladlen Koltun. Monocular visual-inertial depth estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6095–6101, 2023. 8
- [49] Yingye Xin, Xingxing Zuo, Dongyue Lu, and Stefan Leutenegger. Simplemapping: Real-time visual-inertial dense mapping with deep multi-view stereo, 2023. 2
- [50] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [51] 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, 2022. [2](#), [3](#)
- [52] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *Advances in Neural Information Processing Systems*, pages 25018–25032. Curran Associates, Inc., 2022. [5](#)
- [53] 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 (CVPR)*, 2018. [6](#)
- [54] Wei Zhang, Tiecheng Sun, Sen Wang, Qing Cheng, and Norbert Haala. Hi-slam: Monocular real-time dense mapping with hybrid implicit fields, 2023. [2](#), [3](#), [6](#), [7](#)
- [55] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. [2](#), [3](#), [6](#), [7](#), [8](#)
- [56] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [57] 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 (CVPR)*, pages 12786–12796, 2022. [2](#), [3](#)
- [58] 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, 2023. [2](#), [3](#), [6](#), [7](#), [8](#)
- [59] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4):1363–1370, 2020. [2](#)
- [60] Xingxing Zuo, Nathaniel Merrill, Wei Li, Yong Liu, Marc Pollefeys, and Guoquan Huang. Codevio: Visual-inertial odometry with learned optimizable dense depth. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14382–14388, 2021. [2](#)