FACTORIZED NEURAL RADIANCE FIELD WITH DEPTH COVARIANCE FUNCTION FOR DENSE RGB MAPPING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing high-quality and real-time dense maps is critical for building the 3D environment for robot sensing and navigation. Recently, Neural Radiance Field (NeRF) has garnered great attention due to its excellent scene representation capacity of the 3D world; therefore, recent works leverage NeRF to learn 3D maps, typically based on RGB-D cameras. However, depth sensors are not always available for all devices, while RGB cameras are cheap and widely applicable. Therefore, we propose to use single RGB input for the scene reconstruction with NeRF, which becomes highly challenging without geometric guidance from depth sensors. Moreover, we cultivate its real-time capability with lightweight implementation. In this paper, we propose **FMapping**, a factorized NeRF mapping framework, allowing for high-quality and real-time reconstruction with only the RGB input. The insight of our method is that depth doesn't experience much change in consecutive RGB frames, thus the geometrical clues can be derived from RGB effectively with well estimated depth priors. In detail, we divide the mapping into 1) the initialization stage and 2) the on-the-fly stage. First, given trackers are not always stable in the initialization stage, we start with a noisy pose input to optimize the mapping. To this end, we exploit geometric consistency between volume rendering and signed distance function in a self-supervised way to capture depth accurately. In the second stage, given relatively short optimization time for real-time performance, we model the depth estimation as a Gaussian process (GP) with a pre-trained cost-effective depth covariance function to promptly infer depth on the condition of previous frames. Meanwhile, the per-pixel depth estimation and its corresponding uncertainty can guide the NeRF sampling process. Hence, we propose to densely allocate sample points within adjustable truncation regions near the surface, and further distribute samples to ones with high uncertainty. This way, we can continue building maps from subsequent poses with stabilized trackers. Experiments demonstrate that our framework outperforms state-of-the-art RGB-based mapping and achieves comparable performance to RGB-D mapping in terms of photometric and geometric accuracy, with real-time depth estimation capability in around 5 Hz with consistent scale.

1 INTRODUCTION

Robot sensing and navigation rely on building high-quality dense maps in real-time, which provides instant feedback on the environment. Such a paradigm offers notable advantages by providing a comprehensive and instant scene reconstruction, beneficial for onboard tasks, such as robot navigation (Temeltas & Kayak, 2008; Fang et al., 2021) and interactive digital applications (Bettens et al., 2020; Sato et al., 2020). Earlier methods, e. g. , Henry et al. (2014); Dai et al. (2017b) are built based on RGB-D cameras, while their explicitly-cached point clouds impose a high requirement for computation and memory, limiting the practical application to resource-restricted mobile devices.

Neural Radiance Fields (NeRFs) (Mildenhall et al., 2020) have recently emerged as a compelling solution to the mapping problem for 3D reconstruction. For example, Sucar et al. (2021) and Zhu et al. (2022) propose to build neural implicit representations. In other words, NeRF utilizes latent representations, such as a multi-layer perceptron (MLP) to implicitly infer density and color of 3D points instead of caching them directly, thereby reducing memory consumption. These methods are not applicable when no depth sensors are available, as they often have difficulty in estimating

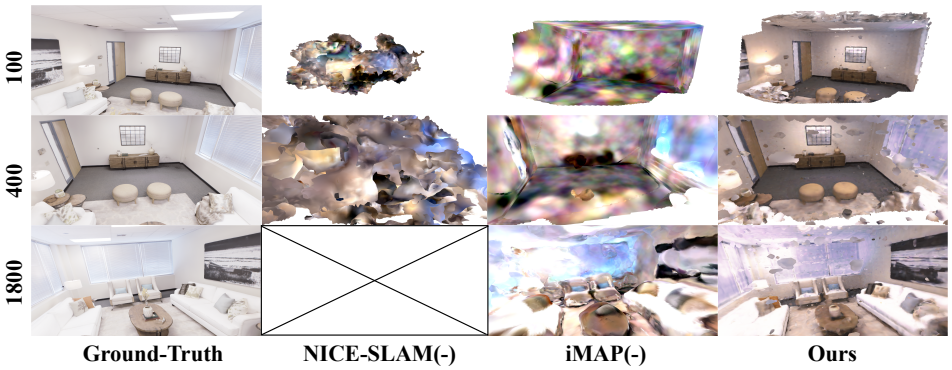


Figure 1: A simple experiment demonstrates the difficulty of real-time scene reconstruction by neural implicit representation without depth supervision. Dense mapping snapshots (at 100, 400, and 1800 input frames) of the on-the-fly running of NICE-SLAM(-) (Zhu et al., 2022), iMAP(-) (Sucar et al., 2021) and our method are displayed for Replica (Room0) sequence, given ground truth (GT) poses without depth supervision. (-) denotes that we make modifications to the original implementations by eliminating the back-propagation of the gradient from depth supervision.

accurate geometric cues. On the other hand, RGB cameras are widely applied with extensive usage, while lack of geometry guidance further poses a challenge on model's convergence during training. Therefore it is valuable to consider real-time mapping with pure RGB input. Some recent NeRF-based methods (Rosinol et al., 2022; Li et al., 2023; Zhu et al., 2023; Chung et al., 2022) have made similar efforts to tackle dense RGB mapping given more geometric constraints. For instance, Rosinol et al. (2022) incorporates geometric cues, e. g. , point clouds, derived from external SLAM systems (Teed & Deng, 2021). Li et al. (2023) performs multi-scale grid occupancy estimation using a cross-frame photometric warping loss. However, these methods require extensive computation to obtain extra geometrical evidence to reach a similar quality from the RGB-D SLAM system. Most of them can not be real-time without customized CUDA implementations.

Besides computation burden, mapping brings unique difficulties, which can be categorized into two primary aspects. (1) **Unstable trackers for pose initialization**: Facing the unknown environment, as observed by Cheng et al. (2021), trackers often do not perform well and have slow convergence speed. Therefore, it would be more realistic to learn mapping from the scratch without poses from trackers. (2) **Slow mapping within limited time**: During on-the-fly stage equipped with trackers, it requires a mapping system adaptable to growing scenes in real-time, while capable of rendering high-quality map. To our best knowledge, most of existing works scarify their real-time performance for high quality map reconstruction. Recently, H_2-Mapping (Jiang et al., 2023) attempts to solve the same issue by proposing a hybrid representation that combines octrees and implicit multi-resolution hash encoding to build maps with RGB-D cameras. However, its memory consumption is considerably large given such representation. On the contrary, we tackle to RGB dense mapping problem with lightweight Factorized NeRF, achieving comparable mapping quality.

In light of this, we present **FMapping**, an efficient neural field mapping framework that facilitates the continuous estimation of a 3D map for dense RGB mapping in real-time. (**I**) We setup the online mapping with RGB stream as a two stage maximal likelihood problem, including the initialization and the on-the-fly continue learning phases, where the prior aims to learn mapping without poses, and the later to achieve online high quality reconstruction with poses. (**II**) Inspired by Chen et al. (2022), we leverage the factorized neural field to decompose the grid features into a lower-dimensional space, slimming model while ensuring its representation ability. (**III**) We leverage the kernel function (Dexheimer & Davison, 2023) to derive the depth guidance, distributing sample on surfaces with high uncertainty and achieving speedy convergence during on-the-fly stage. To maintain the function's stability, we propose a self-supervised depth training method on Signed Distance Function (SDF) and NeRF depth. Consequently, our solution enables high quality, real-time mapping for dense RGB mapping. We show that our FMapping can reconstruct a high-fidelity dense map more efficiently than existing methods with no poses provide in the initialization. During on-the-fly phase, our method achieves real-time high-fidelity mapping with a standard PyTorch implementation, with its map quality comparable to RGB-D based methods.

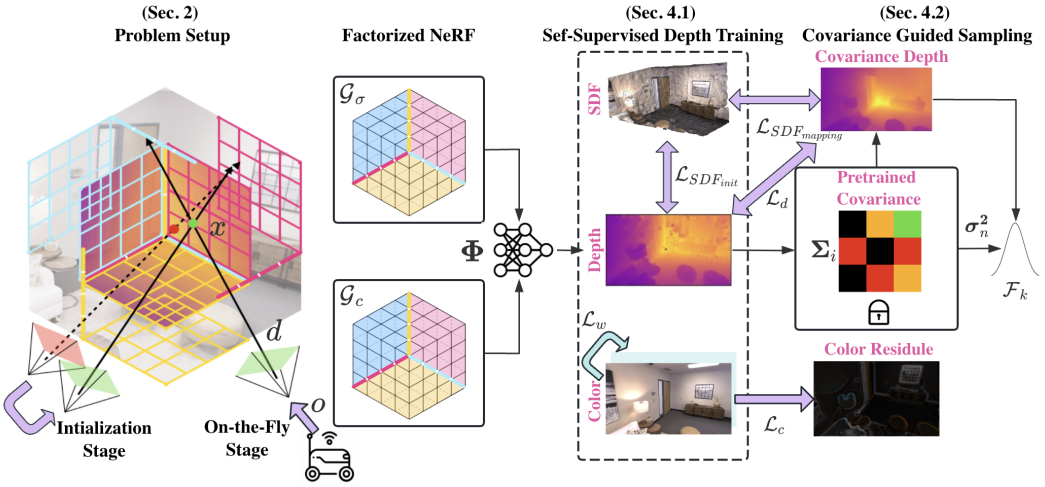


Figure 2: Schematic diagram of the framework. G_σ and G_c represent the geometric and appearance representations, respectively. By decoding representation features, the final color, depth, and SDF estimation are obtained. The entire online training process is constrained by SDF lossesL_{SDF} in a self-supervised way, RGB loss L_c, warpping loss L_w, and covariance depth loss L_d.

2 Problem Setup

We address the challenge of the online instant construction of dense maps with only posed RGB frames as input. In this setup, given image frames $\boldsymbol{I}$ and corresponding poses $\tilde{\boldsymbol{P}}$, we estimate the frames' color $\tilde{\boldsymbol{I}}$ and depth $\tilde{\boldsymbol{D}}$ to reconstruct dense RGB map $\tilde{\boldsymbol{m}}$. We assume that poses obtained from trackers entail a normally distributed disturbance $\boldsymbol{n}$, formulated as $\tilde{\boldsymbol{P}} = \boldsymbol{P} + \boldsymbol{n}, \boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{Q})$. Our goal is instant dense reconstruction upon receiving a posed RGB stream of any accumulated length, i. e. $\boldsymbol{I}_{1:k}$. Inspired by Montemerlo et al. (2002), We formulate the dense mapping problem as estimating a conditional joint probability distribution of:

$$P(\tilde{\boldsymbol{m}}|\tilde{\boldsymbol{P}}_{1:k}, \boldsymbol{I}_{1:k}). \tag{1}$$

NeRF (Mildenhall et al., 2020) has been introduced as a compelling solution for implicit scene representation. Recent works (Chan et al., 2021; Chen et al., 2022; Johari et al., 2022) leverage the matrix decomposition to speed up NeRF computation, i. e. , representing the high-dimensional features by samples' 3D coordinates along with their latent feature, which has emerged as a prevailing technique for NeRF acceleration. Denote NeRF function as $\mathcal{G}$ and its decoder as Φ, the dense mapping problem of 1 is estimating the map $\tilde{\boldsymbol{m}} = \Phi(\mathcal{G}(\boldsymbol{r}))$ to maximize the posterior probability:

$$\tilde{\boldsymbol{m}} = \arg\max_{(\Phi, \mathcal{G}, \tilde{\boldsymbol{r}})} P(\tilde{\boldsymbol{P}}_{(k-w):k}, \boldsymbol{I}_{(k-w):k} | \Phi(\mathcal{G}(\tilde{\boldsymbol{r}}_{(k-w):k}))), \tag{2}$$

where $\tilde{\boldsymbol{r}}(t) = \tilde{\boldsymbol{o}} + t\tilde{\boldsymbol{d}}$ are samples drawn from camera rays originated from the center $\tilde{\boldsymbol{o}}$ with normalized direction $\tilde{\boldsymbol{d}}$, t denotes the camera distance from $\tilde{\boldsymbol{o}}$ to any point sample at $\tilde{\boldsymbol{r}}$. In practice, only partial frame observation is cached in a window of size w to achieve real-time operation.

As shown at Fig. 2, to best align our solutions to the real-world setting, e. g. , unstable trackers, we make two assumptions. **1) Noisy start** with large $\boldsymbol{Q}$ dominates the system optimization during the kick-off of mapping. **2) Stable continuous learning** with mapping uncertainty $\boldsymbol{\Sigma}$ takes major role and has limited noise $\boldsymbol{n}$. Specifically, it can be divided into **Initialization stage** which needs to predict $\tilde{\boldsymbol{I}}$, $\tilde{\boldsymbol{D}}$ and $\tilde{\boldsymbol{P}}$; and the **On-the-fly mapping stage** that estimates $\tilde{\boldsymbol{I}}$, $\tilde{\boldsymbol{D}}$ upon the system's initialization with stabilized pose stream. Therefore, we can view the implicit map construction as a maximal likelihood problem, which is equivalent to minimizing its quadratic form:

$$\begin{aligned} &\arg\min_{(\Phi, \mathcal{G}, \tilde{\boldsymbol{r}}, \tilde{\boldsymbol{P}})} (\tilde{\boldsymbol{I}}_{init} - \boldsymbol{I}_{0:w})^T \boldsymbol{Q}^{-1} (\tilde{\boldsymbol{I}}_{init} - \boldsymbol{I}_{0:w}), \\ &\arg\min_{(\Phi, \mathcal{G}, \tilde{\boldsymbol{r}})} (\tilde{\boldsymbol{I}} - \boldsymbol{I}_{(k-w):k})^T \boldsymbol{\Sigma}^{-1} (\tilde{\boldsymbol{I}} - \boldsymbol{I}_{(k-w):k}). \end{aligned} \tag{3}$$

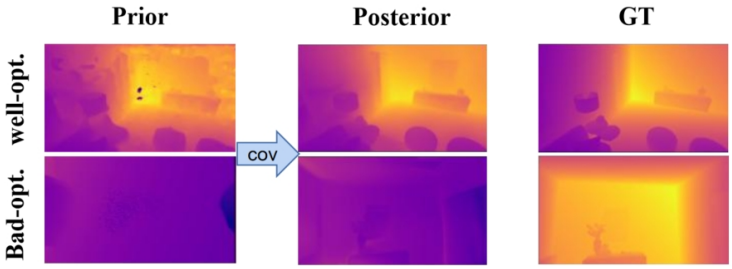


Figure 3: Visual demonstration of covariance depth estimation based on well-optimized neural implicit representation and bad-optimized.

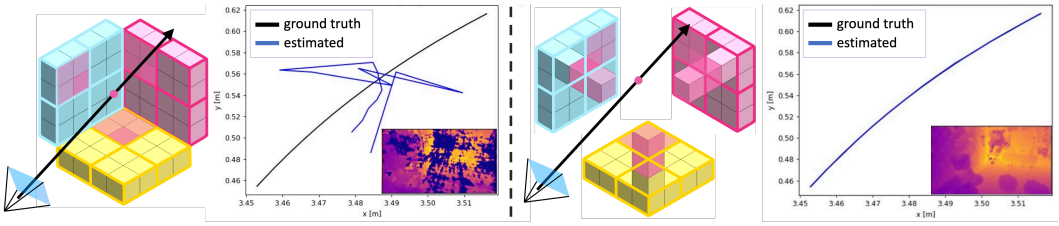
The subsequent issue is how to design a system that constrains the two-stage uncertainty of the system under specified conditions. Given the large window size (Li et al., 2023) or provided depth (Bian et al., 2023), the neural implicit representations can be decently optimized with extremely noisy pose inputs or even no pose inputs. Specifically, during the initialization of the system, these methods use cross-frame consistent constraint to minimize the pose variance $\boldsymbol{Q}$ given a relatively large window size. However, for continuous instant mapping, the large sliding window size is not an optimal choice, since it is hard to converge within limited optimization iterations. Inspired by Dexheimer & Davison (2023), we want to explicitly constrain the covariance $\boldsymbol{\Sigma}$ in Eq. 3 by a cost-effective pre-trained kernel function. The idea is to forecast the correlation between the depths of any two pixels upon receiving an RGB frame. In this way, a covariance function can be constructed to model depth distribution of the current frame, i. e. , we can obtain strong depth prior over RGB frames by modelling the depth function f over any pixels $\boldsymbol{x}$ and $\boldsymbol{x}'$ as a Gaussian Process ($\mathcal{GP}$): $f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$. In this way, pre-trained pixel-wise covariance can be leveraged to infer kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ to provide highly reliable depth priors. Conditioning on them, we proposed a covariance-guided sampling in 4.2 to stabilize instant reconstruction.

Despite of the effectiveness of covariance functions, it highly depends on the well-estimated depth distribution of previous frames, as showed in Fig. 3. Therefore, it's crucial to maintain a relatively accurate depth to guide covariance function to infer depths without self-supervised training described in Sec. 4.1.

3 Related Works

Dense Visual SLAM. Dense visual SLAM has experienced rapid evolution in the past two decades. Compared to sparse visual SLAM algorithms (Klein & Murray, 2007; Mur-Artal & Tardós, 2017) that reconstruct sparse point clouds, dense visual SLAM algorithms (Newcombe et al., 2011b) are able to recover dense point cloud representations of the scene. Some iconic traditional dense SLAM works (Newcombe et al., 2011a; Keller et al., 2013) explicitly represent surfaces using standard volume representation. In addition, some works Dai et al. (2017b); Vespa et al. (2018) employ hierarchical volume representations, which offer increased efficiency but present challenges in implementation and parameter optimization due to their size. Recently, deep learning-based works Czarnowski et al. (2020); Li et al. (2020; 2018) have made great advances in the dense visual SLAM, bringing the benefits of both accuracy improvement and robustness enhancement. *Different from the aforementioned explicit representation methods, we focus on implicitly representing the scene given the posed RGB images, which is compact and can be extended to unobserved regions.*

Monocular depth estimation. With no depth inputs, it is necessary to introduce additional signals to supervise the depth estimation (Zhu et al., 2023). For example, NICER-SLAM employs the off-the-shelf depth estimation model (Eftekhar et al., 2021) to generate the depth ground truth. However, monocular depth estimation is an ill-posed problem due to its scale ambiguity (Bhoi, 2019). To compensate for it, NICER-SLAM adds a scale item to the depth loss. However, it relies on a heavy depth estimation model to ensure the quality of depth ground truth. In addition, the scale problem interferes with the consistency between frames. NICER-SLAM and DIM-SLAM introduce warping loss to enforce the geometric consistency between frames. However, the warping loss only supervises the colour information, instead of directly targeting the structural consistency. *Differently, we leverage the kernel function to derive the depth guidance, which can improve the scale consistency of depths between frames, while maintaining the lightweight and real-time advantages.*



Pseudo Tri-Plane (Tensor Approximation) **Factorization (Tensor Decomposition)**

Figure 4: Heuristic comparison of two efficient approximations of hybrid representation of Neural Radiance Filed, i. e. , Tri-projection (**Left**) and Factorization (**Right**) in the system initialization stage with only RGB supervision, which is a joint estimation of initial poses and implicit maps. The pose trajectory plot and the depth estimation suggest that the factorized 4D tensors are more robust to elusive camera trajectory and unobserved geometry.

Implicit Dense Mapping. Implicit representations have demonstrated their capacity to encode scenes within a latent feature space through the utilization of a single Multi-Layer Perceptron (MLP). The implicit representation has manifested in various applications across diverse domains, including novel view synthesis (Mildenhall et al., 2020; Müller et al., 2022; Verbin et al., 2022; Zhang et al., 2020), surface reconstruction (Yariv et al., 2021; Oechsle et al., 2021; Wang et al., 2021), as well as the creation and manipulation of scenes and avatars (Liu et al., 2021; Yang et al., 2021).

Neural Radiance Fields (NeRF) serve as an example that generates novel views based on sparse input data through a single MLP. It has spurred subsequent research about dense mapping. Pioneering this effort, iMAP (Sucar et al., 2021) demonstrated that a single MLP can effectively represent a 3D scene, even extending to unobserved regions. To extend the implicit dense mapping to larger scenes, NICE-SLAM (Zhu et al., 2022) employs hierarchical voxel grids and pre-trained decoders to enhance the representation capacity. Additionally, ESLAM Johari et al. (2022) replaces the voxel grids utilized in NICE-SLAM with compact feature planes, improving the speed significantly.

To reduce the demand for depth inputs, NeRF-SLAM (Rosinol et al., 2022) and Orbeez-SLAM (Chung et al., 2022) have been integrated explicitly the NeRF into visual SLAM systems. However, this integration results in redundant architectures. On the other hand, NICER-SLAM (Zhu et al., 2023) relies on heavy pre-trained geometric models, which can not meet real-time demand. Recently, DIM-SLAM (Li et al., 2023) introduced the first dense RGB SLAM system entirely based on the neural implicit mapping. However, the problem has been downgraded into estimating voxel grid occupancy using a single channel without exploiting the robustness and expressiveness of high-dimensional latent features. Additionally, such occupancy estimation requires tri-linear interpolation of stacked grids of many different resolutions, leading to an undesirable computational budget.

4 FMAPPING

4.1 SELF-SUPERVISED DEPTH TRAINING

As depicted at Fig. 2, we denote factorized neural radiance field as 4D representation $\mathcal{G}$ and propose a self-supervised strategy that leverages geometric consistency to speed up the mapping speed. The first goal is to improve efficiency. It can be achieved by decomposing NeRF feature computation in $\mathcal{G}$ by multiplying the matrix and vector in the lower dimensional space. Recent works Chan et al. (2021); Chen et al. (2022); Johari et al. (2022) leverage the matrix decomposition to speed up NeRF computation, i. e. , projecting the high-dimensional features to their low dimensional counterparts, which has emerged as a prevailing technique for NeRF acceleration. In Fig. 4, we evaluate the two common decomposition paradigms, namely the Tri-plane representation that projects the 3D tensor onto three 2D feature planes; and the BTD-based Factorized representation (Chen et al., 2022) that interprets it as multiplication between matrix and vectors. The trajectory and depth estimation results Fig. 4 show that the Factorization scheme appears to be more robust during the initialization stage. Therefore, we leverage NeRF factorization to enhance computational efficiency without dampening the rendering fidelity. To cultivate the potential of representation ability of $\mathcal{G}$ in mapping with RGB, it is a common practice to leverage geometric consistency derived from images. For instance,

Li et al. (2023) proposes to enforce geometric consistency using a warping loss $\mathcal{L}_w$ in the spirit of multi-view stereo (Zheng et al., 2014). We follow the same paradigm in Li et al. (2023) by enforcing multi-scale geometric loss:

$$\begin{aligned} \mathcal{L}_w &= \frac{1}{\mathcal{M}} \sum_{\boldsymbol{q}_j \in \mathcal{M}} \sum_{j,l} \sum_{s \in \mathcal{S}} \boldsymbol{B}_{\boldsymbol{q}_j} SSIM(\mathcal{N}^s_{\boldsymbol{q}_j}, \mathcal{N}^s_{\boldsymbol{q}_{j\to l}}), j \neq l, \\ \boldsymbol{q}_{j\to l} &= \boldsymbol{K}_l \tilde{\boldsymbol{R}}_l^{\,T} (\tilde{\boldsymbol{R}}_j \boldsymbol{K}_j^{-1} \boldsymbol{q}_j^h \tilde{D}_{\boldsymbol{q}_j} + \tilde{\boldsymbol{T}}_j - \tilde{\boldsymbol{T}}_l), \end{aligned} \tag{4}$$

where randomly sampled $|\mathcal{M}|$ pixels and their corresponding patch $|\mathcal{N}|$ sizes $\mathcal{S}$ from cached frames inside the initialization window are re-projected to neighbouring frames. Next, Reflecting structural similarity ($SSIM$) is applied to calculate the difference inside the visibility mask $\boldsymbol{B}$. The 2D pixel $\boldsymbol{q}_j$ in frame j is lifted into 3D space and then project it to another frame l represented by $\boldsymbol{q}_{j\to l}$. $\tilde{\boldsymbol{P}}$ is the estimated poses, and $\boldsymbol{q}_j^h$ is the homogeneous coordinates of $\boldsymbol{q}_j$, $\boldsymbol{K}$ is the camera intrinsic.

4.2 Covariance Guided Sampling

Given NeRF function $\mathcal{G}$ and its decoder Φ, jointly denoted as parameters θ, density and colors are estimated by underlying continuous volumetric scene function $\sigma_\theta(\tilde{\boldsymbol{x}})$ and $c_\theta(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}})$. Volume rendering (Mildenhall et al., 2020) aims to enhance spatial coherence by integrating estimated samples $\tilde{\boldsymbol{x}}$ along rays $\tilde{\boldsymbol{r}}$ for color supervision,

$$\begin{aligned} \tilde{\boldsymbol{I}}(\tilde{\boldsymbol{r}}) &= \sum_{i=1}^{N} \alpha_\theta(\tilde{\boldsymbol{x}}_i) \prod_{n<i} (1 - \alpha_\theta(\tilde{\mathbf{x}_n}))\, c_\theta(\tilde{\mathbf{x}_i}, \tilde{\boldsymbol{d}}), \\ \alpha_\theta(\tilde{\boldsymbol{x}}_i) &= 1 - exp(-\sigma_\theta(\tilde{\boldsymbol{x}}_i)\delta_i), \end{aligned} \tag{5}$$

where $\alpha_\theta(\tilde{\boldsymbol{x}}_i)$ denotes the penetrating light at $\tilde{\boldsymbol{x}}_i$ and then composites the sample radiance into the rendered frames. Therefore, following the conventions, e. g. Zhu et al. (2022), the depth $\tilde{\boldsymbol{D}}_k$ and color $\tilde{\boldsymbol{I}}_k$ of pixel can be formulated as:

$$\tilde{\boldsymbol{D}}_k = \sum_{i=1}^{N} w_\theta(\tilde{\boldsymbol{x}}_i) t_i,\ \tilde{\boldsymbol{I}}_k = \sum_{i=1}^{N} w_\theta(\tilde{\boldsymbol{x}}_i) c_\theta(\tilde{\mathbf{x}_i}, \tilde{\boldsymbol{d}}), \tag{6}$$

where $w_\theta(\tilde{\boldsymbol{x}}_i) = \alpha_\theta(\tilde{\boldsymbol{x}}_i) \prod_{n<i} (1 - \alpha_\theta(\tilde{\mathbf{x}_j}))$. $\alpha_\theta(\tilde{\boldsymbol{x}}_i)$(Eq. 5) entails uncertainty regarding sample distribution δ. Oechsle et al. (2021) uses the sigmoid activation directly on $\sigma_\theta(\tilde{\boldsymbol{x}}_i)$ to avoid the ambiguity, i. e. , $\alpha_\theta(\tilde{\boldsymbol{x}}_i)$ is replaced by $O_\theta(\tilde{\boldsymbol{x}}_i) = Sigmoid(\sigma_\theta(\tilde{\boldsymbol{x}}_i))$ in Eq. 5.

However, as shown in Fig. 2, without direct sensor depth supervision like Sucar et al. (2021); Zhu et al. (2022); Johari et al. (2022), $\tilde{\boldsymbol{D}}_k$ is indirectly constrained through color rendering loss $\mathcal{L}_c = ||\tilde{\boldsymbol{I}}_k - \boldsymbol{I}_k||_2^2$. Inspired by recent works on instant reconstruction (Johari et al., 2022; Wang et al., 2023) which leverages Signed Distance Function (SDF) to build the underlying geometry. We designed implicit mapping to infer depth based on the rendering equation while simultaneously outputting SDF Fig. 2. Since camera distances, i. e. , depth values are equal regarding SDF and inferred depth from NeRF, it allows for self-supervision to put pixel-wise depth constraint to extend further to spatial occupancy coherence:

$$\mathcal{L}_{SDF}(\boldsymbol{\omega}) = \frac{1}{\mathcal{K}} \sum_{k \in \mathcal{K}} \frac{1}{\mathcal{P}} \sum_{i \in \mathcal{P}} \Big(t_i + \Phi_{\boldsymbol{g}}(\mathcal{G}(\boldsymbol{x}_i)) \cdot Tr \cdot \boldsymbol{\omega} - \tilde{\boldsymbol{D}}_k) \Big)^2, \tag{7}$$

where $\Phi_{\boldsymbol{g}}(\mathcal{G}(\boldsymbol{x}_i))$ gives SDF value estimation of sampler $\boldsymbol{x}_i$, based on its distance t_i start from a camera origin and truncation distance Tr on any ray. ω is weights to distribute samples according to underlying uncertainty. We set $\boldsymbol{\omega} = 1$ for $\mathcal{L}_{SDF_{init}}$ loss in the initialization stage. Furthermore, we can also use $\mathcal{L}_d = ||\tilde{\boldsymbol{D}}_k - \boldsymbol{D}_{k_*}||_2^2$ to supervise depth derived from NeRF.

At the on-the-fly stage, we assume both pose variance and representation are stabilized. As discussed in Sec. 2, we can utilize the pre-trained covariance function to infer depth maps upon receiving the newest input images. In detail, condition on geometric estimations made by neural implicit functions, we gain depths prior from predicted depth distribution (Dexheimer & Davison, 2023):

$$\mathbf{f}_* \sim \mathcal{N}\left(m(\tilde{\boldsymbol{D}}_{j\to l}), K(\boldsymbol{I}_l, \boldsymbol{I}_l)\right). \tag{8}$$

Upon receiving $\boldsymbol{I}_l$, we calculate the mean $m(\cdot)$ of the re-projected depth $\tilde{\boldsymbol{D}}_{j\to l}$ from the last frame , which is available after using photometric warping from the last frames. Specifically, the posterior distribution of the depth function of frame l can be calculated to obtain its predictive covariance depth $\tilde{\boldsymbol{D}}_{l_*}$ and covariance Σ_{l_*}:

$$\begin{aligned}\tilde{\boldsymbol{D}}_{l_*} &= m(\tilde{\boldsymbol{D}}_{j\to l}) + K_{\mathrm{fn}}(K_{\mathrm{nn}} + \sigma_n^2 I)^{-1}(\tilde{\boldsymbol{D}}_{j\to l} - m(\tilde{\boldsymbol{D}}_{j\to l})),\\ \Sigma_{l_*} &= K_{\mathrm{ff}} - K_{\mathrm{fn}}(K_{\mathrm{mn}} + \sigma_n^2 I)^{-1}K_{\mathrm{nf}},\end{aligned} \tag{9}$$

where K stands for positive semi-definite matrix provided by covariance function given n samples. σ_n^2 capture per-pixel depth estimation uncertainty and thus can be cached inside an optimization window pending to supervise the sampling procedure jointly with the $\tilde{\boldsymbol{D}}_{l_*}$ for instant neural implicit mapping. Intuitively speaking, as we assumed the existence of well estimated priors, σ_n^2 is reliable to highlight complex regions to explore in the later phase, relaxing biased surface region which entails large uncertainty for better estimation. Therefore, we weights pixels in an element-wise way by $\boldsymbol{\omega} = \sigma_n^2$ in Eq. 7 for $\mathcal{L}_{SDF_{mapping}}$ loss in the on-the-fly stage.

At last, as shown in Fig. 2, we adjust the sampling distribution to consider high-reward regions around the objects' surface, i. e. , indicated by reliable covariance depth $\tilde{\boldsymbol{D}}_{l_*}$. During NeRF's rendering process, N samples $X_r = \{\tilde{\boldsymbol{x}}_i | \tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{r}}(t_i), t_i < t_{i+1}\}_{i=1}^N$ along any estimated ray $\tilde{\boldsymbol{r}}$ are drawn from the coarse stratified sampling, followed by the inverse transform sampling according to the coarse-level sampling $\mathcal{F}_{cdf}(\tilde{\boldsymbol{x}})$ over its normalized PDF scores $\alpha_\theta(\tilde{\boldsymbol{x}})$,

$$X_{r,k+1} = \mathcal{F}_{pdf}^{-1}(u) \cup X_{r,k},\ \mathcal{F}_{cdf}(\boldsymbol{x}_{i,\tilde{k+1}}) = \sum_i P(\boldsymbol{x}_{i,k}^{\tilde{}} | \alpha_\theta(\boldsymbol{x}_{i,k}^{\tilde{}}) < u), u \in \boldsymbol{U}, \tag{10}$$

where $\boldsymbol{U} \sim Unif[0,1]$, and k denotes the iteration times for multi-stage estimation, e. g. , $k = 2$ in the coarse-to-fine hierarchical sampling. The resulting adjacent sample distance is $\delta_i = |\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|$ from Eq. 5. $X_{r,k}$ is sorted according to their camera distances after each sampling iteration.

Specifically, as illustrated in Fig. 2, the iterative sampling process consists of a truncation sampling $\mathcal{F}_{tr}$ over the k_{th} round samples with camera distances $\boldsymbol{t}_{r,k}$. Similar to the SDF loss defined in Eq. 7, $\mathcal{F}_{tr}$ aims to sample points close to $\tilde{\boldsymbol{D}}_{l_*}$, i. e. , the surface regions for the geometric constraint. To obtain fine level of granularity, we guide $\mathcal{F}_{tr}$ with uncertainty map $\boldsymbol{\sigma}_n^2$ of each depth map $\tilde{\boldsymbol{D}}_{l_*}$. In detail, we enlarge the truncation intervals $Tr \cdot \boldsymbol{\sigma}_n^2$ for rays corresponding to pixel regions with higher uncertainty. Then we integrate samples with the previous round, and the inverse transform sampling result $\mathcal{F}_{cdf}$ from Eq. 10 for more robust sample estimation.

$$\begin{aligned}X_{r,k+1} &= \mathcal{F}_{cdf}^{-1}(\boldsymbol{U}) \cup \mathcal{F}_{tr}(\boldsymbol{t}_{r,k}) \cup X_{r,k},\\ \mathcal{F}_{tr}(\boldsymbol{t}_{r,k}) &= \tilde{\boldsymbol{o}} + \boldsymbol{D}_{l_*} + \boldsymbol{t}_{r,k} \cdot Tr \cdot \boldsymbol{\sigma}_n^2.\end{aligned} \tag{11}$$

5 EXPERIMENTS

We provide an evaluation of our FMapping on simulated dataset (Straub et al., 2019) quantitatively and qualitatively against the common real-time neural implicit scene reconstruction benchmarks, including the RGB-D method, e. g. , iMAP (Sucar et al., 2021), NICE-SLAM (Zhu et al., 2022) and H2-Mapping (Jiang et al., 2023); and RGB method like Orbeez-SLAM (Chung et al., 2022). We

also included the experimental results of the NICE-SLAM running without depth supervision that is available in Rosinol et al. (2022).

5.1 EXPERIMENTAL SETTING

Covariance guided sampling: At each inference, the multi-stage sampling with $k = 4$ is performed, where the stratified samples are collected at the first iteration for rough distribution estimation, followed by 3 iterations of covariance guided sampling $\mathcal{F}_{tr}$. During each inference, we use the updated uncertainty map to setup truncation intervals with a max value of The sampling sizes of each stage are 32, 64, 64, and 64, respectively. For each sampling iteration, 40% samples are first retrieved from $\mathcal{F}_{cdf}^{-1}$, then the remaining 60% are picked through $\mathcal{F}_{tr}$.

The initialization phase: We collect 15 frames for jointly estimating initial poses and local implicit maps. Once the initialization stage is finished, the first frame is added to the global keyframe set and kept fixed. The total loss during initialization phase is denoted as $\mathcal{L}_{init} = \beta_c\mathcal{L}_c + \beta_d\mathcal{L}_d + +\beta_w\mathcal{L}_{w,s\in\{1,5,11\}} + \beta_s\mathcal{L}_{SDF_{init}}$, where β represents a weighting factor.

The on-the-fly mapping phase: In the process of mapping, we maintain an active window of 20 frames, with the portion of the global and local frame the same with Li et al. (2023). 20 iterations of optimization are performed to update the map for every 5 frames. The oldest 5 local frames are removed while the new 5 incoming frames are added to the window for the next map update. we also leverage the uncertainty guidance in Eq. 11 on balancing wrapping loss $\mathcal{L}_{w,s=1}$ and SDF loss $\mathcal{L}_{SDF_{mapping}}$ besides sampling procedure. The total loss during on-the-fly stage is denoted as $\mathcal{L}_{fly} = \beta_c\mathcal{L}_c + \beta_d\mathcal{L}_d + (\beta_w\mathcal{L}_{w,s=1} + \beta_s\mathcal{L}_{SDF_{mapping}})|\mathcal{P} \otimes \boldsymbol{\sigma}_k^2$, where $\otimes$ is the element wise multiplication operation that weights each patch with normalized uncertainty map before using them for loss calculation.

Evaluation Metrics. We assess the precision in terms of both geometric and photometric quality. To measure geometric accuracy, we rely on the L1 depth error, which compares the estimated and ground-truth depth maps. Note that we follow the common practice to recover the metric scale by aligning the median of the estimated depth with the ground truth, as also used in Bian et al. (2023); Zhou et al. (2017). For photometric accuracy, we use the peak signal-to-noise ratio (PSNR) to analyze the similarity between the input RGB images and the rendered images.

Implementation Details. All experiments are conducted on a single NVIDIA RTX 3090 GPU. The factorized representation is inspired and implemented based on the TensoRF (Chen et al., 2022) and the pre-trained depth covariance function is made available by Dexheimer & Davison (2023), which has been trained on Scannet Dataset (Dai et al., 2017a). The resolution of the factorized feature grid is computed based on the pre-defined bounding box. We implement a single-resolution factorized feature grid with a dimension set to 64. To make it comparable with existing neural implicit mapping methods using a voxel grid, the resolution is roughly calculated as a voxel size of $\sim$ 8cm, given a bounding box of size 11.8m, 8.7m, and 6.8m for three coordinates (the example is given for Replica scene room 0). The feature channels are set to 16 for both density and appearance components, respectively. Both SDF and color decoders are a two-layer MLP that explains the appearance feature. Adam optimizer (Kingma & Ba, 2014) is adopted with learning rates set to 0.02 for the grid feature updating and set to 0.001 for color decoder updating, respectively. Note that some benchmark scene e. g. *office1*, is dimming and thus lacks color variance that poses difficulty in constraining SDF, so the SDF supervision is muted for better leverage the available appearance feature. A covariance depth and its corresponding uncertainty is estimated for the incoming downsampled RGB image (set to 2.5 in the Replica case) for efficient inference. Three consecutive cached sets of RGB images, the covariance depth map and the uncertainty map are sent to our Fmapping with an additional 20 global sampled overlapped frames to jointly optimize the neural implicit representation for 30 iterations.

5.2 RESULTS

As shown in Tab. 1, our method demonstrates generally better geometric and photometric estimation results compared to other RGB instant mapping cases and even shows comparable performance to the state-of-the-art RGB-D mapping methods (H2-mapping). Note that we report the depth output from depth covariance (cov) inferring and neural implicit rendering (rend), respectively.

Table 1: Quantitative comparison of our proposed method's mapping performance on Replica indoor scenes.

Method		room0	room2	office0	office1	office2	office3	office4	Avg.
iMAP (RGB-D)	Depth L1 ↓	5.70	6.94	6.43	7.41	14.23	8.68	6.80	7.64
	PSNR. ↑	5.66	5.64	7.39	11.89	8.12	5.62	5.98	6.95
NICE-SLAM (RGB-D)	Depth L1 ↓	2.53	2.93	1.51	0.93	8.41	10.48	2.43	4.08
	PSNR. ↑	**29.90**	19.80	22.44	25.22	22.79	22.94	24.72	24.61
H2-mapping (RGB-D)	Depth L1 ↓	**0.34**	**0.61**	**0.33**	**0.45**	**0.53**	**0.50**	**0.40**	**0.42**
	PSNR. ↑	29.24	**27.05**	**33.72**	**33.82**	**28.91**	**29.43**	**31.17**	**30.21**
NICE-SLAM (RGB*)	Depth L1 ↓	11.12	19.03	11.12	10.24	16.36	21.33	14.81	14.18
	PSNR. ↑	18.15	17.82	20.23	19.14	15.22	**16.12**	17.24	17.76
Orbeez-SLAM (RGB)	Depth L1 ↓	-	-	-	-	-	-	-	11.88
	PSNR. ↑	-	-	-	-	-	-	-	**29.25**
FMapping (RGB)	Depth L1 ↓ (cov)	**0.21**	**0.51**	0.16	**0.29**	**0.40**	**0.96**	0.32	**0.41**
	Depth L1 ↓ (rend)	**0.21**	0.60	**0.15**	0.30	0.42	1.00	**0.30**	0.43
	PSNR. ↑	**24.32**	**26.03**	**30.20**	**36.49**	**27.26**	16.08	**24.94**	26.47

* Note that the result of the *room1* is omitted here since the initialization stage does not generate satisfactory prior to kicking off the following Gaussian process.

Table 2: Analysis of our method in comparison with existing ones in terms of mapping speed, number of parameters, and model size growth rate (parameterized by scene side-length L).

Method	Mapping Speed ↓ [s]	Memory ↓ # Param.	Grow. R.
iMAP (RGB-D)	0.45	0.22 M	-
NICE-SLAM (RGB-D)	**0.13**	12.18 M	$O(L^3)$
Ours-cov (RGB)	0.19	∼36.00 M	-
Ours-rend (RGB)	2.40	**0.025 M**	$O(L^2)$

In Tab. 2, we compare our mapping speed against common real-time RGB-D neural implicit method, i. e. iMAP (Sucar et al., 2021) and NICE-SLAM (Zhu et al., 2022). Our FMapping can achieve comparable online estimation speed. Due to the lack of absolute sensor depth, an additional dedicated dynamic sampling process is required for FMapping to approach the true geometry compared to our RGB-D counterparts, therefore resulting in more processing time. Finally, regarding memory consumption, our representation is memory efficient, We ascribe it to the factorized neural field representation. Despite our covariance depth estimator based on a pre-trained covariance function entailing a large parameter size, it is a relatively cheap geometric prior with a real-time inference capability and naturally possesses the capability of cross-frame consistency for real-time reconstruction tasks, compared to other works that leverage large pre-trained monocular depth estimator (Zhu et al., 2023).

6 Conclusion

In this paper, we present FMapping, an efficient neural field mapping technique for real-time dense RGB mapping. We leverage a light and flexible geometric prior, i.e., a depth covariance function, to continuously estimate depth based on well-optimized neural implicit mapping upon receiving RGB observations. In return, this supervises the online training of NeRF. We leverage factorized neural field representation to facilitate fast convergence with efficient memory growth. We achieve state-of-the-art RGB mapping in terms of photometric and geometric accuracy, and our results are even comparable to the performance of RGB-D dense mapping.

REFERENCES

Anne M Bettens, Benjamin Morrell, Mauricio Coen, Neil McHenry, Xiaofeng Wu, Peter Gibbens, and Gregory Chamitoff. Unrealnavigation: Simulation software for testing slam in virtual reality. In *AIAA Scitech 2020 Forum*, pp. 1343, 2020.

Amlaan Bhoi. Monocular depth estimation: A survey. *arXiv preprint arXiv:1901.09402*, 2019.

Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. 2023.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, 2022.

Jun Cheng, Liyan Zhang, and Qihong Chen. An improved initialization method for monocular visual-inertial slam. *Electronics*, 10(24), 2021. ISSN 2079-9292. doi: 10.3390/electronics10243063. URL https://www.mdpi.com/2079-9292/10/24/3063.

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. *arXiv preprint arXiv:2209.13274*, 2022.

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.

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*, pp. 5828–5839, 2017a.

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 Transactions on Graphics (ToG)*, 36(4):1, 2017b.

Eric Dexheimer and Andrew J. Davison. Learning a depth covariance function. 2023.

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*, pp. 10786–10796, 2021.

Baofu Fang, Gaofei Mei, Xiaohui Yuan, Le Wang, Zaijun Wang, and Junyang Wang. Visual slam for robot navigation in healthcare facility. *Pattern recognition*, 113:107822, 2021.

Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Experimental robotics: The 12th international symposium on experimental robotics*, pp. 477–491. Springer, 2014.

Chenxing Jiang, Han-Qi Zhang, Peize Liu, Zehuan Yu, Hui Cheng, Boyu Zhou, and Shaojie Shen. H$_{2}$-mapping: Real-time dense mapping using hierarchical hybrid representation. *IEEE Robotics and Automation Letters*, 8:6787–6794, 2023. URL https://api.semanticscholar.org/CorpusID:259089291.

Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: efficient dense SLAM system based on hybrid representation of signed distance fields. *CoRR*, abs/2211.11704, 2022. doi: 10.48550/arXiv.2211.11704. URL https://doi.org/10.48550/arXiv.2211.11704.

Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pp. 1–8. IEEE, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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*, pp. 225–234. IEEE Computer Society, 2007. doi: 10.1109/ISMAR.2007.4538852. URL https://doi.org/10.1109/ISMAR.2007.4538852.

Heng Li, Xiaodong Gu, Weihao Yuan, Luwei Yang, Zilong Dong, and Ping Tan. Dense rgb slam with neural implicit maps. *ArXiv*, abs/2301.08930, 2023.

Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7286–7291. IEEE, 2018.

Ruihao Li, Sen Wang, and Dongbing Gu. Deepslam: A robust monocular slam system with unsupervised deep learning. *IEEE Transactions on Industrial Electronics*, 68(4):3577–3587, 2020.

Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.

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 *ECCV*, 2020.

Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103. URL https://doi.org/10.1109/TRO.2017.2705103.

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*, pp. 127–136. Ieee, 2011a.

Richard A. Newcombe, Steven Lovegrove, and Andrew J. Davison. DTAM: dense tracking and mapping in real-time. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc Van Gool (eds.), *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pp. 2320–2327. IEEE Computer Society, 2011b. doi: 10.1109/ICCV.2011.6126513. URL https://doi.org/10.1109/ICCV.2011.6126513.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5589–5599, 2021.

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.

Yuki Sato, Kojiro Minemoto, Makoto Nemoto, and Tatsuo Torii. Construction of virtual reality system for radiation working environment reproduced by gamma-ray imagers combined with slam technologies. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 976:164286, 2020.

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.

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*, pp. 6209–6218. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00617. URL https://doi.org/10.1109/ICCV48922.2021.00617.

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.

Hakan Temeltas and Demiz Kayak. Slam for robot navigation. *IEEE Aerospace and Electronic Systems Magazine*, 23(12):16–19, 2008.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5481–5490. IEEE, 2022.

Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul HJ Kelly, and Stefan Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, 2018.

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13293–13302, 2023.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13779–13788, 2021.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1510–1517, 2014. doi: 10.1109/CVPR.2014.196.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1851–1858, 2017.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 12776–12786. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01245. URL https://doi.org/10.1109/CVPR52688.2022.01245.

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. *CoRR*, abs/2302.03594, 2023. doi: 10.48550/arXiv.2302.03594. URL https://doi.org/10.48550/arXiv.2302.03594.