Swift-Mapping: Online Neural Implicit Dense Mapping in Urban Scenes

Ke Wu[1], **Kaizhao Zhang**[2], **Mingzhe Gao**[3], **Jieru Zhao**[3], **Zhongxue Gan**[1*], **Wenchao Ding**[1* †]

[1]Academy for Engineering & Technology, Fudan University
[2]School of Future Technology, Harbin Institute of Technology
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University
23110860017@m.fudan.edu.cn, 8200880120@stu.hit.edu.cn, {a823337391z, zhao-jieru}@sjtu.edu.cn,
{ganzhongxue, dingwenchao}@fudan.edu.cn

Abstract

Online dense mapping of urban scenes is of paramount importance for scene understanding of autonomous navigation. Traditional online dense mapping methods fuse sensor measurements (vision, lidar, etc.) across time and space via explicit geometric correspondence. Recently, NeRF-based methods have proved the superiority of neural implicit representations by high-fidelity reconstruction of large-scale city scenes. However, it remains an open problem how to integrate powerful neural implicit representations into online dense mapping. Existing methods are restricted to constrained indoor environments and are too computationally expensive to meet online requirements. To this end, we propose Swift-Mapping, an online neural implicit dense mapping framework in urban scenes. We introduce a novel neural implicit octomap (NIO) structure that provides efficient neural representation for large and dynamic urban scenes while retaining online update capability. Based on that, we propose an online neural dense mapping framework that effectively manages and updates neural octree voxel features. Our approach achieves SOTA reconstruction accuracy while being more than 10x faster in reconstruction speed, demonstrating the superior performance of our method in both accuracy and efficiency.

Introduction

For autonomous navigation in open urban scenes, it is essential to comprehend complex urban environments with high accuracy and efficiency to meet the navigation requirements. Online dense mapping, which involves the real-time construction and updating of detailed scene representations, has emerged as a critical component for achieving this level of scene understanding.

In the field of online dense mapping, conventional methods have heavily relied on explicit geometric matching to fuse sensor measurements over time and space. Techniques like multi-view stereo and simultaneous localization and mapping (SLAM) have been the cornerstones of these approaches (Newcombe, Lovegrove, and Davison 2011; Koestler et al. 2022; Lin and Zhang 2022). However, these methods often struggle when dealing with the complex and ever-changing nature of urban scenes, like efficiency and memory challenges, rapid ego motion, occlusions, and scale variations.

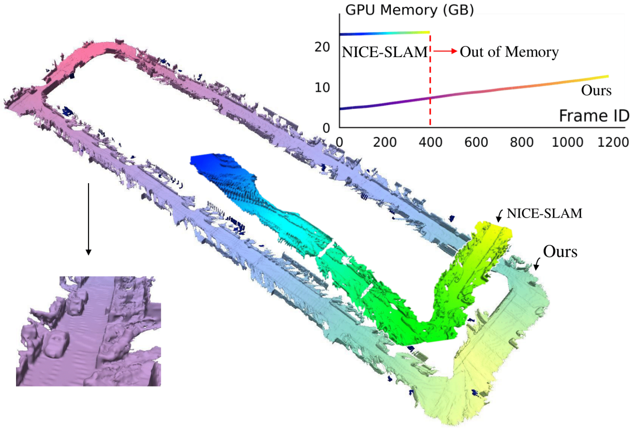

Figure 1: 3D reconstruction of urban scenes (KITTI) using Swift-Mapping. Swift-mapping provides a high-fidelity and accurate reconstruction for scene details while being capable of scaling to large scale city scenes utilizing the neural implicit octomap structure (NIO).

In contrast, recent advances in neural implicit representations, as exemplified by the pioneering NeRF-based techniques (Mildenhall et al. 2021), have showcased remarkable potential in reconstructing expansive urban landscapes using naive and lightweight network structures and simple losses. For instance, NeRF-W (Martin-Brualla et al. 2021) introduces an innovative approach by combining Multilayer Perceptrons (MLP) with latent vectors, enhancing NeRF's adaptability across diverse scenes and facilitating the reconstruction of high-quality city-scale environments. Block-NeRF (Tancik et al. 2022) decomposes an urban scene into independent MLPs by spatial location and composites multiple MLPs during inference. However, it is worth noting that these methods primarily emphasize the fidelity of reconstruction, often embracing intricate models and resource-intensive training processes. Integrating neural implicit rep-

[*]Corresponding authors: Zhongxue Gan and Wenchao Ding.

[†]This work is sponsored by Shanghai Pujiang Program (23PJ1400900).

resentations into *online* dense mapping for urban scenes remains an ongoing challenge.

iMAP (Sucar et al. 2021) is a pioneering effort in integrating neural implicit representations into online SLAM systems. This method involves transforming a 3D query point into an occupancy or color value using a single Multilayer Perceptron (MLP) and optimizing this MLP through volume rendering. In contrast, NICE-SLAM (Zhu et al. 2022) takes a different route by utilizing uniform dense feature grids for scene representation. This strategy provides a structured approach for seamlessly integrating multi-level scene intricacies, resulting in enhanced efficiency and robustness. However, NICE-SLAM's applicability is confined to controlled indoor environments due to its dense feature representation and localized voxel updates. *To the best of our knowledge, there is a noticeable absence of online dense mapping methods specifically tailored for urban scenes that fully leverage the potential of neural implicit representations.*

Current approaches encounter dual challenges, concerning both efficiency and accuracy when dealing with extensive urban scenes. Dense voxel-based methods like NICE-SLAM struggle with maintaining high-resolution dense feature voxels for superior reconstruction quality. While this ensures high fidelity, it hampers scalability and efficiency in urban scenarios. On the other hand, Keyframe-based approaches like iMAP face the challenge of optimizing a substantial number of keyframes simultaneously. This complexity can become burdensome when applied to expansive urban environments.

To confront the challenges of both efficiency and accuracy, we introduce Swift-Mapping, an innovative framework for online neural implicit dense mapping in urban settings. Our approach encompasses a novel feature representation known as the neural implicit octomap (NIO). This octree voxel grid not only facilitates efficient neural representation for expansive and dynamic urban scenes but also preserves the capability for real-time updates. Building upon this foundation, we establish an online neural dense mapping framework that continuously update neural octree voxel features. A working example is shown in Fig. 1.

Our contributions can be summarized as follows:

- A novel neural implicit octomap (NIO) that provides efficient neural representation for large and dynamic urban scenes while retaining online update capability.
- An online dense mapping framework that effectively manages and updates neural octree voxel features, which advances the frontier of online, high-fidelity scene understanding in dynamic outdoor urban settings.
- Extensive experiments compared to SOTA methods, which validate the superior performance of our framework, with higher accuracy while being orders of magnitude faster in terms of reconstruction speed.

Related Works

Implicit Dense Mapping

Neural Radiance Fields (NeRF) have gained substantial traction in the realm of realistic mapping. In the NeRF framework (Mildenhall et al. 2021), a singular 5D coordinate (x, y, z, θ, ϕ) is mapped to a volume density and its view-dependent RGB color through a trainable Multilayer Perceptron (MLP). This framework has been extended across numerous subsequent studies (Bi et al. 2020a; Rudnev et al. 2022; Bi et al. 2020b; Zhang et al. 2021; Zeng et al. 2023), encompassing tasks such as relighting and handling sparse input views (Niemeyer et al. 2022; Roessle et al. 2022; Rebain et al. 2022), and more.

Nonetheless, a notable constraint persists concerning computational time. The NeRF framework is primarily tailored for offline reconstruction and novel view rendering. It is not directly applicable to the domain of online dense mapping. To overcome this efficiency challenge, a slew of voxel-based NeRF variants have been devised (Yu et al. 2021; Zhu et al. 2022; Sun, Sun, and Chen 2022; Yang et al. 2022; Zhang et al. 2022; Li et al. 2022; Jiang et al. 2023; Takikawa et al. 2021). For instance, NVSF (Liu et al. 2020) introduces a set of sparse voxel octree-structured implicit fields to enhance temporal efficiency and scalability. However, NVSF remains confined to offline applications and lacks the capability for incremental and on-the-fly updates.

NICE-SLAM (Zhu et al. 2022), on the other hand, merges a dense feature grid with an MLP decoder to boost efficiency. Yet, it's important to note that NICE-SLAM is largely restricted to indoor environments due to the constraints posed by dense feature voxels.

In light of these challenges, our paper proposes a neural implicit octomap (NIO). NIO not only facilitates photorealistic rendering but also exhibits the potential for seamless scalability to vast urban scenes, all while permitting incremental expansion in an online fashion. This innovation paves the way for unlocking new avenues for online, high-fidelity urban scene understanding and mapping.

Urban Scale Mapping

Urban scale mapping remains a formidable challenge due to the vastness, dynamic elements, and inherent difficulty in acquiring depth information within such environments. Nonetheless, there have been efforts to deploy NeRF in urban contexts. For instance, (Rematas et al. 2022) leverages lidar data to supervise geometry learning. Meanwhile, READ (Li, Li, and Zhu 2023) employs a rendering network known as w-net to glean neural descriptors from sparse data. DNMP (Lu et al. 2023) introduces a neural variant of the traditional mesh representation. S-NeRF (Xie et al. 2023) simultaneously addresses novel view synthesis for both urban scenes and moving vehicles in the foreground. Notably, both (Li, Li, and Zhu 2023) and (Lu et al. 2023) exhibit scene editing capabilities.

However, despite these strides, the inherent complexity of their network architectures inhibits swift convergence and rapid urban reconstruction. Thus, even with the accomplishments achieved by (Li, Li, and Zhu 2023; Lu et al. 2023), the pursuit of achieving fast convergence and efficient urban reconstruction in an online manner remains a standing challenge.

In this paper, we introduce a sparse hybrid sampling strategy and hierarchical latent vectors based on NIO, which

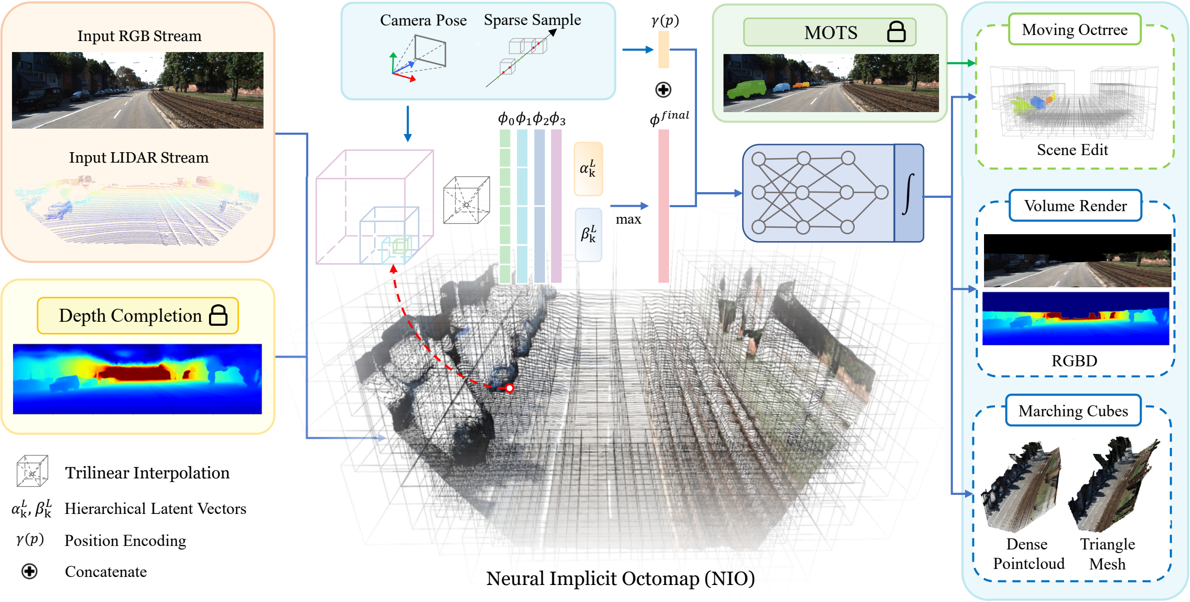

Figure 2: The Swift-Mapping pipeline utilizes RGB and LiDAR streams along with pose data as inputs, and incrementally builds the neural implicit octomap (NIO) structure. Valid octree voxels are associated with latent features. Hierarchical latent vectors can be queried from NIO, which are fed to MLP decoders for volume rendering. The feature vectors of NIO and MLP decoders can be supervised through simple photometric loss. Furthermore, the framework has another side product. It is capable of modeling moving obstacles and even achieves scene editing through manipulating the feature voxels of NIO.

enable incremental update of NIO while preserving high-fidelity photo-realistic scene understanding.

Method

We present an overview of our method in Fig. 2. Initially, RGB-D or RGB-SparseD (LIDAR) stream is fed into our system. We employ the neural implicit octomap (NIO) as the scene representation to conduct incremental spatial and temporal fusion on input streams. During online dense mapping, we utilize a NeRF-like update scheme. For each sample ray, we extract hierarchical latent vectors from NIO and train two MLP decoders for occupancy and color prediction. Moreover, our framework supports volume rendering, 3D mesh extraction, and dynamic obstacle modeling.

Neural Implicit Octomap

The NIO is essentially a neural hierarchical voxel map based on an octree structure. We incorporate two types of features in NIO, namely, depth feature Φ_ω^d and color feature Φ_ω^c. For each type of feature, we use a separate MLP with learnable parameters to represent the geometrical and photometric information, respectively.

Octree-based Voxel Grid Traditional methods like NICE-SLAM adopt dense uniform voxel grids, which scale poorly in urban scenes. In this paper, we employ a hierarchical voxel grid where feature voxels are organized using octrees, as depicted in Fig. 2. We denote the depth of the octree as K and the resolution of the smallest voxel as l. Readers may refer to (Hornung et al. 2013) for details of octrees.

With the point clouds produced by a camera pose and the corresponding depth map, we allocate voxels based on the octree structure. To reduce the impact of sensor measurements, a voxel at the k-th level is marked as valid only if more than two $k-1$-th level subvoxels are valid. Invalid voxels are filtered and no longer processed. This strategy helps avoid excessive memory consumption due to outliers. The octree-based voxel grid is built **incrementally**. In the continuous mapping process, if a point is related to a particular valid voxel that already exists, no operation is performed. If no invalid voxel is related, we activate the voxel at that position and randomly initialize a latent feature vector, adding it to the octree structure.

We find that the octree structure facilitates information propagation between consecutive frames, especially under fast ego motion. When using dense feature grids, totally different voxels may be sampled for adjacent frames, which is problematic for fusing temporal information in dense mapping. Leveraging the octree structure, it is much easier to sample related voxels (possibly at different levels) between adjacent frames, and spatial and temporal information is fused through the expansion of the octree.

Distance Adaptive Voxel Initialization Traditional methods like NICE-SLAM adopt the same resolution for nearby regions and distant areas. For indoor scenes, this scheme is acceptable since the scale variation is small. However, for large urban scenes, there exist large scale variations. For NeRF-based optimization, a ray through a particular pixel in the image plane will affect distinct volumes for nearby regions and distant areas. Using a global coarse voxel resolution will hinder the fidelity of nearby regions while a global fine-grained resolution will pose unacceptable memory requirements. What is worse, using fine-grained resolution for distant areas may hinder the spatial temporal information fusion since correlated samples are scattered across different fine-grained voxels due to imperfect depth measurements.

In NIO, there are K different voxel resolutions for different levels of octree. When allocating a new voxel for a particular point, the voxel resolution is chosen based on the scheme that nearby regions use dense small voxels, while distant areas adopt sparse large voxels. Suppose the maximum sampling distance is $d_{\max}$ which corresponds to the minimum resolution $l \cdot 2^{K-1}$. The voxel resolution is calculated in line with the scale variation at different distances. The voxel resolution for level $k \in \{0, 1, ..., K-1\}$ covers points sampled from distance $2^{k-K} \cdot d_{\max}$ to $2^{k-K+1} \cdot d_{\max}$. More details are provided in the supplementary materials.

Dense Mapping Using Hierarchical Latent Vectors

In the traditional methods, keyframe selection plays a pivotal role in ensuring long-term map consistency and preventing catastrophic forgetting (Yang et al. 2022). However, for large urban scenes, a large number of keyframes are required to prevent forgetting, which results in a highly inefficient optimization process, especially when using neural implicit representation. Instead of maintaining a keyframe list, we propose using hierarchical latent vectors to learn the scene appearance and geometry. As introduced in NIO, each voxel is associated with a latent feature vector.

Sparse Hybrid Voxel Sampling Traditional SLAM methods using neural implicit representations often require RGBD inputs. However, for online dense mapping, depth information is often noisy. Especially, for urban scenes, perfect depth is hard to obtain. To this end, we do not assume perfect depth is always available. Our method can cope with sparse depth input (i.e., from LIDAR measurements) instead of dense RGBD. We adopt CompletionFormer (Zhang et al. 2023) depth completion to interpolate the sparse depth to dense depth, which facilitates the sampling process in NeRF-style update. Note that completed depth maps often contain noise.

Due to the noise in depth input, simply sticking to surface sampling is insufficient. To this end, we propose using a hybrid sampling strategy, namely, combining random sampling with surface sampling. As a result, more accurate depth values for a certain spatial point are more likely to be sampled. Given a single frame, we perform hybrid sampling based on the camera pose T_i and the imperfect depth map D_i. Along each ray, we compute surface samples N_s points from the depth map, and randomly sample N_r points, resulting in a total of $N = N_s + N_r$ points.

Due to the sparse nature of NIO, not every point can be associated with a valid grid. Instead of directly computing the intersection between a sampled ray and NIO which can be computationally expensive, we adopt a sparse point-wise sampling strategy which is compatible with NIO. Specifically, we randomly sample a large number of points along each ray and check their validity. Note that checking whether a point falls into a valid voxel is a simple division operation and is extremely efficient. For points in invalid voxels, we skip the subsequent hierarchical latent vector extraction and MLP update and use zero padding for volume rendering. This operation will significantly reduce computational costs for volume rendering.

Hierarchical Latent Vector Extraction For point $\mathbf{p}$ which is successfully associated with a valid voxel, we can extract hierarchical latent vectors from relevant voxels up to K levels (incl. all super voxels). For a particular level k, we extract two fused latent vectors, namely, $\phi_k^d(\mathbf{p})$ for depth feature and $\phi_k^c(\mathbf{p})$ for color feature. The latent vectors are extracted using tri-linear interpolation among eight neighboring voxels at the same level. For the sub-voxels of lower levels which are not associated with this point, a dummy zero-filled latent vector filled is adopted.

After acquiring hierarchical latent vectors, the fusion of features from various levels becomes necessary. For NIO, we design the dimension of the feature vector to scale with the level. The dimension of the latent vector at level k is twice that of level $k-1$. According to this design, for each level k, we form the expanded latent vector ϕ_k^{depth} and ϕ_k^{color} as follows:

$$\begin{aligned} \phi_k^{\text{depth}}(\mathbf{p}) &= (\phi_k^d(\mathbf{p}), \alpha_k \phi_k^d(\mathbf{p}), ..., \alpha_k^{2^{K-k}} \phi_k^d(\mathbf{p})) \\ \phi_k^{\text{color}}(\mathbf{p}) &= (\beta_k^c(\mathbf{p}), \beta_k \phi_k^c(\mathbf{p}), ..., \beta_k^{2^{K-k}} \phi_k^c(\mathbf{p})) \\ \phi^d(\mathbf{p}) &= \max_k(|\phi_k^d(\mathbf{p})|), \phi^c(\mathbf{p}) = \max_k(|\phi_k^c(\mathbf{p})|) \end{aligned} \tag{1}$$

where α_k and β_k denotes learnable memorization parameters. Each frame is associated with learnable α_k and β_k which controls the memorization and forgetting of NIO. We adopt max pooling operator to fuse the concatenated latent vectors from different levels.

MLP Decoder The point $\mathbf{p}$ is first processed through position encoding. Then, the encoded position representation is concatenated with $\phi^d(\mathbf{p})$ or $\phi^c(\mathbf{p})$, and fed into two separated MLPs f_θ^{d} and f_ω^{c} to decode occupancy probability and color value:

$$o_{\mathbf{p}} = f_\theta^{\text{d}}(\mathbf{p}, \phi^d(\mathbf{p})), \quad c_{\mathbf{p}} = f_\omega^{\text{c}}(\mathbf{p}, \phi^c(\mathbf{p})). \tag{2}$$

For the network structure of f_θ^{d} and f_ω^{c}, we follow the design of the ConvOnet (Peng et al. 2020) decoder, and use 5 fully-connected blocks, with residual connections added for the third block. θ and ω denote the trainable parameters of the two MLPs. Based on a series of decoded density $o_{\mathbf{p}}$ and color $c_{\mathbf{p}}$ along a sampled ray, volume rendering is conducted. We refer interested readers to (Mildenhall et al. 2021) for details of volume rendering. Finally, for each ray, depth value $\hat{D}_m$ and RGB color $\hat{I}_m$ are estimated. The predicted results are then compared with the ground truth depth D_m and RGB values I_m to calculate the L2 loss:

$$\begin{aligned} \mathcal{L}_d &= \frac{1}{M} \sum_{m=1}^{M} (D_m - \hat{D}_m)^2 \\ \mathcal{L}_p &= \frac{1}{M} \sum_{m=1}^{M} (I_m - \hat{I}_m)^2 \end{aligned} \tag{3}$$

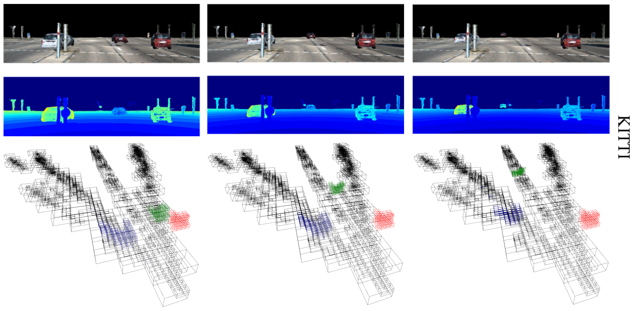

Figure 3: NIO not only supports modeling dynamic obstacles, and also supports scene editing functionality by manipulating the octree feature grid as depicted above. We highlight that this feature is highly promising for fast simulation from sensor measurements.

where $\mathcal{L}_d$ denotes the geometric loss and $\mathcal{L}_p$ denotes the photometric loss. Finally, we optimize the trainable parameters $\{\theta, \omega, \alpha, \beta, \phi^d, \phi^c\}$ as follows:

$$\min_{\theta,\omega,\alpha,\beta,\phi^d,\phi^c} (\lambda_d \mathcal{L}_d + \lambda_p \mathcal{L}_p) \tag{4}$$

Modeling Dynamic Urban Scenes

Moving Octree NIO is capable of modeling dynamic obstacles. We implement MOTSFusion(Luiten, Fischer, and Leibe 2020) to obtain the semantic labels for each vehicle in each frame. Suppose we need to customize the i-th vehicle according to the pose sequence $\{\tau_i^t\}$ within time sequence $\{t\}$. We first train NIO and freeze all the features and MLPs. Then we project the depth of the i-th vehicle into world coordinate to calculate corresponding point clouds P_i and get the voxels V_i belonging to the i-th vehicle as follows:

$$\begin{aligned} P_i^t = \tau_i^t P_i^{t-1}, V_i^t &= \left\lfloor \frac{P_i^t - 0}{l \cdot 2^k} \right\rfloor \\ V_i^t - V_i^{t-1} &\approx \left\lfloor \frac{(\tau_i^t - I)P_i^{t-1}}{l \cdot 2^k} \right\rfloor \end{aligned} \tag{5}$$

V_i^t represents the voxel index of the i-th vehicle and P_i^t represents the point cloud. We reconstruct dynamic vehicles by moving vehicle's voxels and corresponding features on the octree structure. We only move fine-level voxels, which avoids affecting coarse-level background voxels. Meanwhile, benefiting from the sparse structure, only a few static voxels are occluded and replaced by moving voxels.Different future scenes can be rendered at very low cost without computationally expensive training. The interesting part is that we can further achieve scene editing functionality based on the NIO structure.

Experiments

Experiment Settings

Datasets We conduct experiments on both indoor and urban datasets. For indoor testing, we employ the synthetic Replica dataset (Straub et al. 2019) and the real-world ScanNet dataset (Dai et al. 2017). We also include datasets containing diverse urban scenes such as KITTI (Geiger et al. 2013), VKITTI2 (Cabon, Murray, and Humenberger 2020), and nuScenes (Caesar et al. 2019).

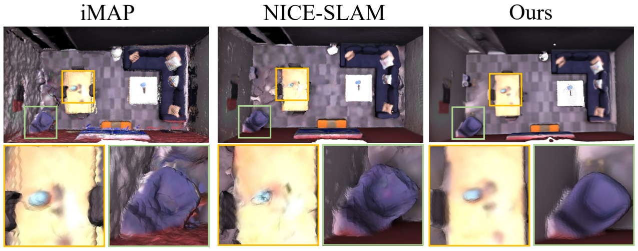

Figure 4: Reconstruction results on the Replica Dataset.

Baselines We conduct comparisons against two sophisticated NeRF-based SLAM methods, namely iMAP (Sucar et al. 2021) and NICE-SLAM (Zhu et al. 2022), both of which employ RGB-D images as inputs. Since the scope of this paper is dense mapping, we feed ground truth poses to all the methods. Additionally, we compare our method against two offline NeRF-based reconstruction methods, Instant-NGP and Mip-NeRF-360 (Müller et al. 2022; Barron et al. 2022). Note that Instant-NGP and Mip-NeRF-360 are both computationally expensive and not targeted for online applications. We include them to serve as strong baselines in terms of reconstruction accuracy.

Metrics We evaluate both the accuracy and efficiency of all the methods. For 2D metrics, we employ Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE). For 3D metrics, we include Accuracy (Acc), Completion Ratio (Comp Ratio $<$ 5cm %). Due to space limitations, interested readers may find detailed descriptions for the metrics in (Zhu et al. 2022). For urban datasets (KITTI, VKITTI2, nuScenes), due to the lack of ground truth meshes, only 2D metrics are evaluated.

Evaluation

Indoor Reconstruction For indoor experiments on Replica dataset, we follow the settings of NICE-SLAM. Since indoor datasets have ground truth 3D meshes, 3D metrics are calculated. A qualitative example of the reconstructed 3D mesh is shown in Fig. 4. Although our method is suitable for large urban scenes, it still has comparable performance against NICE-SLAM which is primarily designed for indoor scenes. Quantitative results are shown in Table 1. Thanks to the sparsity nature of NIO, our method has a higher convergence speed and much lower memory usage. Reaching comparable accuracy, our method only consumes one-third memory of NICE-SLAM.

Urban Scene Rendering For KITTI and nuScenes (both including real-world urban scenes), due to the unknown depth in the sky region, we masked out the rendering for the sky region. In Table 2, we evaluate two 2D metrics, L2 Loss and PSNR, with rendered images shown in Fig. 5. For urban

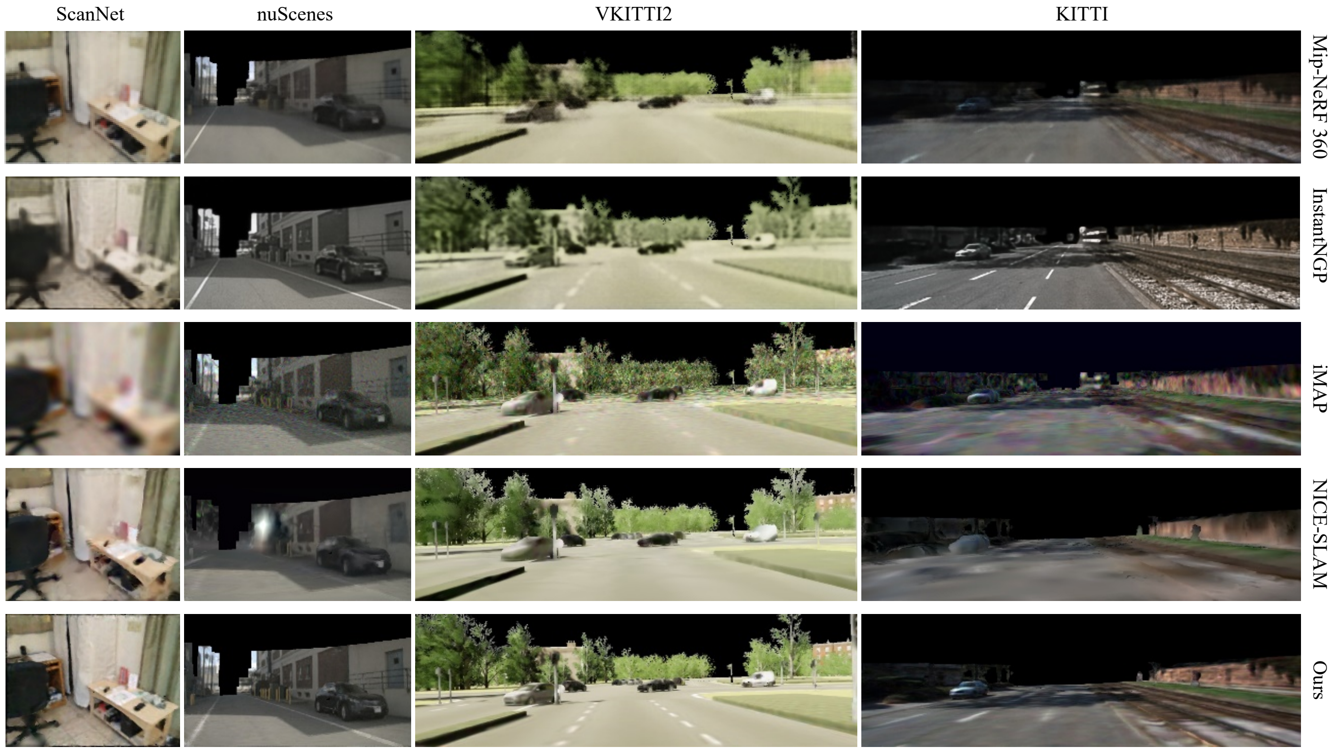

Figure 5: Volume rendering results on the ScanNet (Dai et al. 2017), nuScenes (Caesar et al. 2019), VKITTI2 (Cabon, Murray, and Humenberger 2020), and KITTI (Geiger et al. 2013) datasets. We compare our method with two online methods (NICE-SLAM (Zhu et al. 2022), iMAP (Sucar et al. 2021) and two offline methods (Mip-NeRF-360 (Barron et al. 2022) and Instant-NGP (Müller et al. 2022)). The two offline methods take 10 minutes for training.

	iMAP	NICE-SLAM	Ours
Acc.↓	6.9872	1.7510	**1.6219**
Comp. Ratio↑	62.5970	**86.0861**	84.4212
FPS↑	0.11	0.45	**0.71**

Table 1: Reconstruction results for the Replica Dataset (average over 8 scenes). FPS is evaluated on a single RTX3090.

scenes, both Mip-NeRF (Barron et al. 2022) and Instant-NGP (Müller et al. 2022) take 10 minutes for training on a single NVIDIA RTX3090. It is worth noting that, even compared to offline methods, our method still offers superior rendering quality while being at least 10x faster. Compared to iMAP and NICE-SLAM these online methods, our approach also outperforms them significantly in urban scenes.

Performance Analysis

Using NIO structure instead of using keyframe list is an important design choice for method. The motivation of using keyframe list is to avoid catastrophic forgetting. In this section, we show that the NIO structure can effectively avoid catastrophic forgetting while being scalable and sparse. We further show the robustness of NIO under fast ego motion.

Avoiding Catastrophic Forgetting To assess the effect of forgetting, we propose a PSNR metric PSNR_{mem} which is given by $\text{PSNR}_{\text{mem}} = \text{PSNR}(I_{\text{train}}, I_{\text{infer}})$, where I_{train} denotes the rendered image during training and I_{infer} denotes the rendered image during inference. Higher PSNR_{mem} represents less information is forgotten after optimizing NIO and MLP decoders, which indicates a higher capability of remembering scene appearance and geometry. The detailed results are shown in Table 3. We can find that our method achieves higher $\text{PSNR}_{\text{train}}$, $\text{PSNR}_{\text{infer}}$ and PSNR_{mem} in urban scenes, which validates the superiority of using NIO for avoiding catastrophic forgetting.

Fast Ego Motion Unlike indoor scenes, there may be high speed ego movement in urban scenes. Fast ego motion causes a smaller overlap between adjacent frames, which poses challenges for spatial and temporal information fusion. To validate the performance of different methods under fast ego motion, we uniformly sample frames using different sampling frequencies from original datasets (ScanNet and VKITTI2), which approximates different speeds of ego-motion. As shown in Fig 7, for higher ego-motion speed, the accuracy of NICE-SLAM drops quickly, especially for large-scale outdoor environments. In contrast, the accuracy of our method is stable, indicating the robustness of our method against fast ego motion.

Ablation Study

	ScanNet(indoor)		Replica(indoor)		VKITTI2(urban)		nuScenes(urban)		KITTI(urban)		
	L2 Loss↓	PSNR↑	L2 Loss↓	PSNR↑	L2 Loss↓	PSNR↑	L2 Loss↓	PSNR↑	L2 Loss↓	PSNR↑	FPS↑
iMAP	547.43	20.45	566.25	20.71	637.92	20.02	681.40	20.07	1022.54	18.03	0.02
NICE-SLAM	340.47	22.81	179.51	25.59	930.32	19.15	483.95	21.25	993.18	18.25	0.05
Mip-NeRF	720.08	19.59	432.68	21.76	687.59	19.91	727.14	19.54	847.13	19.68	-
Instant-NGP	669.74	19.80	187.07	24.04	566.44	20.87	306.74	22.95	516.54	21.09	-
Ours	**245.19**	**24.23**	**88.32**	**28.67**	**307.10**	**23.12**	**290.66**	**23.15**	**359.68**	**22.78**	**0.14**

Table 2: Quantitative evaluation of volume rendering on two indoor datasets (Dai et al. 2017; Straub et al. 2019) and three urban datasets (Caesar et al. 2019; Cabon, Murray, and Humenberger 2020; Geiger et al. 2013).

Method	NICE-SLAM		Ours	
Dataset	ScanNet	VKITTI2	ScanNet	VKITTI2
$\text{PSNR}_{\text{train}}$ ↑	21.86	18.21	23.76	22.84
$\text{PSNR}_{\text{infer}}$ ↑	22.32	14.54	21.96	21.95
PSNR_{mem} ↑	22.61	17.25	21.37	23.29

Table 3: Comparison between NICE-SLAM (Zhu et al. 2022) using a keyframe list and our method using hierarchical latent vectors.

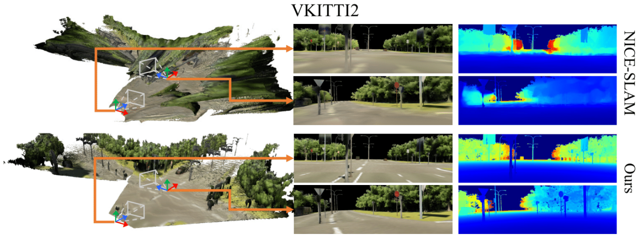

Figure 6: Qualitative results on VKITTI2.

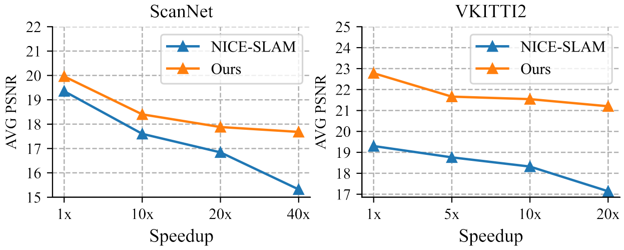

Figure 7: Comparison of PSNR for different sampling intervals (representing different movement speeds).

Octree Depth We validate the performance of our method by varying the octree depth while maintaining the same resolution for the smallest voxel. We control the number of iterations to be the same (i.e., 10 iterations) and compare the efficiency, accuracy under different octree depth setups. As shown in Table 4, deeper octree results in higher accuracy in limited iterations, indicating better convergence.

Level	1	2	3	4	5
FPS↑	0.46	0.39	0.36	0.30	0.26
L2 Loss↓	712.09	535.32	420.03	339.68	317.01
PSNR↑	19.56	20.91	21.95	22.62	23.02

Table 4: Ablative study on using different octree depth.

Dataset	ScanNet		VKITTI2	
α_k, β_k	×	√	×	√
$\text{PSNR}_{\text{train}}$ ↑	23.75	**23.76**	21.91	**22.84**
$\text{PSNR}_{\text{infer}}$ ↑	20.20	**21.96**	18.62	**21.95**
PSNR_{mem} ↑	16.52	**21.37**	17.83	**23.29**

Table 5: Ablation study on α_k, β_k.

Trainable Memorization Parameters α_k, β_k The role of trainable memorization parameters is to control the memorization and forgetting of NIO across multiple frames. To validate this functionality, we replace α_k with an all-one matrix and replace β_k with a matrix drawn from a random normal distribution to examine the performance without using α_k, β_k. The results are shown in Table 5, where we can find that by using learnable parameters for memorizing history information, it effectively preventing network forgetting and ensuring map consistency.

Conclusion

We propose Swift-Mapping, a novel online neural implicit dense mapping method in urban scenes. To address challenges of dense mapping in large scale urban scenes, we introduce the neural implicit octomap (NIO) feature representation, which is sparse, efficient and facilitates spatial temporal information fusion. Based on NIO, we propose a neural online dense mapping process based on the hierarchical latent vectors extracted from NIO. Additionally, we showcase that our framework is robust to fast ego-motion, and is capable of modeling dynamic obstacles and providing scene editing capabilities. Through extensive experiments, our method achieves SOTA performance in both efficiency and accuracy.

Acknowledgments

This work was sponsored by the Shanghai Pujiang Program (23PJ1400900) and further supported by the Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). Additionally, I extend my appreciation to the Shanghai Engineering Research Center of AI Robotics, Fudan University, China, and the Engineering Research Center of AI Robotics, Ministry of Education, China, for their invaluable support in the successful completion of this project.

References

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5470–5479.

Bi, S.; Xu, Z.; Srinivasan, P.; Mildenhall, B.; Sunkavalli, K.; Hašan, M.; Hold-Geoffroy, Y.; Kriegman, D.; and Ramamoorthi, R. 2020a. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*.

Bi, S.; Xu, Z.; Sunkavalli, K.; Hašan, M.; Hold-Geoffroy, Y.; Kriegman, D.; and Ramamoorthi, R. 2020b. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 294–311. Springer.

Cabon, Y.; Murray, N.; and Humenberger, M. 2020. Virtual KITTI 2. arXiv:2001.10773.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2019. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*.

Hornung, A.; Wurm, K. M.; Bennewitz, M.; Stachniss, C.; and Burgard, W. 2013. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, 34: 189–206.

Jiang, C.; Zhang, H.; Liu, P.; Yu, Z.; Cheng, H.; Zhou, B.; and Shen, S. 2023. H2-Mapping: Real-time Dense Mapping Using Hierarchical Hybrid Representation. *arXiv preprint arXiv:2306.03207*.

Koestler, L.; Yang, N.; Zeller, N.; and Cremers, D. 2022. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning*, 34–45. PMLR.

Li, H.; Yang, X.; Zhai, H.; Liu, Y.; Bao, H.; and Zhang, G. 2022. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*.

Li, Z.; Li, L.; and Zhu, J. 2023. Read: Large-scale neural scene rendering for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1522–1529.

Lin, J.; and Zhang, F. 2022. R 3 LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, 10672–10678. IEEE.

Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.-S.; and Theobalt, C. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33: 15651–15663.

Lu, F.; Xu, Y.; Chen, G.; Li, H.; Lin, K.-Y.; and Jiang, C. 2023. Urban Radiance Field Representation with Deformable Neural Mesh Primitives. *arXiv preprint arXiv:2307.10776*.

Luiten, J.; Fischer, T.; and Leibe, B. 2020. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2): 1803–1810.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7210–7219.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4): 1–15.

Newcombe, R. A.; Lovegrove, S. J.; and Davison, A. J. 2011. DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, 2320–2327. IEEE.

Niemeyer, M.; Barron, J. T.; Mildenhall, B.; Sajjadi, M. S.; Geiger, A.; and Radwan, N. 2022. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5480–5490.

Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; and Geiger, A. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 523–540. Springer.

Rebain, D.; Matthews, M.; Yi, K. M.; Lagun, D.; and Tagliasacchi, A. 2022. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1558–1567.

Rematas, K.; Liu, A.; Srinivasan, P. P.; Barron, J. T.; Tagliasacchi, A.; Funkhouser, T.; and Ferrari, V. 2022. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12932–12942.

Roessle, B.; Barron, J. T.; Mildenhall, B.; Srinivasan, P. P.; and Nießner, M. 2022. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12892–12901.

Rudnev, V.; Elgharib, M.; Smith, W.; Liu, L.; Golyanik, V.; and Theobalt, C. 2022. NeRF for Outdoor Scene Relighting. In *European Conference on Computer Vision (ECCV)*.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; Clarkson, A.; Yan, M.; Budge, B.; Yan, Y.; Pan, X.; Yon, J.; Zou, Y.; Leon, K.; Carter, N.; Briales, J.; Gillingham, T.; Mueggler, E.; Pesqueira, L.; Savva, M.; Batra, D.; Strasdat, H. M.; Nardi, R. D.; Goesele, M.; Lovegrove, S.; and Newcombe, R. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv preprint arXiv:1906.05797*.

Sucar, E.; Liu, S.; Ortiz, J.; and Davison, A. J. 2021. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6229–6238.

Sun, C.; Sun, M.; and Chen, H.-T. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5459–5469.

Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; and Fidler, S. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes.

Tancik, M.; Casser, V.; Yan, X.; Pradhan, S.; Mildenhall, B.; Srinivasan, P. P.; Barron, J. T.; and Kretzschmar, H. 2022. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8248–8258.

Xie, Z.; Zhang, J.; Li, W.; Zhang, F.; and Zhang, L. 2023. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*.

Yang, X.; Li, H.; Zhai, H.; Ming, Y.; Liu, Y.; and Zhang, G. 2022. Vox-Fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 499–507. IEEE.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5752–5761.

Zeng, C.; Chen, G.; Dong, Y.; Peers, P.; Wu, H.; and Tong, X. 2023. Relighting Neural Radiance Fields with Shadow and Highlight Hints. In *ACM SIGGRAPH 2023 Conference Proceedings*, 1–11.

Zhang, X.; Bi, S.; Sunkavalli, K.; Su, H.; and Xu, Z. 2022. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5449–5458.

Zhang, X.; Srinivasan, P. P.; Deng, B.; Debevec, P.; Freeman, W. T.; and Barron, J. T. 2021. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (ToG)*, 40(6): 1–18.

Zhang, Y.; Guo, X.; Poggi, M.; Zhu, Z.; Huang, G.; and Mattoccia, S. 2023. Completionformer: Depth completion with convolutions and vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18527–18536.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12786–12796.