NeRF-LOAM: Neural Implicit Representation for Large-Scale Incremental LiDAR Odometry and Mapping

Junyuan Deng
Qi Wu
Xieyuanli Chen
Songpengcheng Xia
Zhen Sun
Guoqing Liu
Wenxian Yu
Ling Pei

1Shanghai Key Laboratory of Navigation and Location Based Services, Shanghai Jiao Tong University
2College of Intelligence Science and Technology, National University of Defense Technology

Abstract

Simultaneously odometry and mapping using LiDAR data is an important task for mobile systems to achieve full autonomy in large-scale environments. However, most existing LiDAR-based methods prioritize tracking quality over reconstruction quality. Although the recently developed neural radiance fields (NeRF) have shown promising advances in implicit reconstruction for indoor environments, the problem of simultaneous odometry and mapping for large-scale scenarios using incremental LiDAR data remains unexplored. To bridge this gap, in this paper, we propose a novel NeRF-based LiDAR odometry and mapping approach, NeRF-LOAM, consisting of three modules neural odometry, neural mapping, and mesh reconstruction. All these modules utilize our proposed neural signed distance function, which separates LiDAR points into ground and non-ground points to reduce Z-axis drift, optimizes odometry and voxel embeddings concurrently, and in the end generates dense smooth mesh maps of the environment. Moreover, this joint optimization allows our NeRF-LOAM to be pre-trained free and exhibit strong generalization abilities when applied to different environments. Extensive evaluations on three publicly available datasets demonstrate that our approach achieves state-of-the-art odometry and mapping performance, as well as a strong generalization in large-scale environments utilizing LiDAR data. Furthermore, we perform multiple ablation studies to validate the effectiveness of our network design. The implementation of our approach will be made available at https://github.com/JunyuanDeng/NeRF-LOAM.

1. Introduction

Simultaneous odometry and mapping is an important component for autonomous mobile systems to achieve full autonomy in large-scale environments. It estimates the 6-degree-of-freedom poses of the vehicle and simultaneously builds a map of the environment, which are fundamental prerequisites for downstream tasks like path planning and collision avoidance. LiDAR sensors have been widely adopted for odometry and mapping due to their ability to provide precise range measurements and robustness to illumination changes. However, it can be argued that the current LiDAR odometry and mapping algorithms prioritize tracking quality over dense reconstruction quality, which may overlook the potential benefits of accurately capturing environmental geometry and generating high-fidelity reconstructions. Despite the popularity of LiDAR-based incremental pose estimation [15, 41, 26, 39], research on high-level dense map reconstruction, especially deep-learning-based algorithms remains scarce.

Recently, neural radiance fields (NeRF) [32] has shown promising potentials in representing 3D scenes implicitly using a neural network and parallelly pose tracking methods [33, 51, 45]. Although such representation can achieve seminal reconstruction with accurate poses, they concentrate on indoor pose tracking and scene representation with RGB-D sensors. The sparsity of LiDAR data and the lack of RGB information present significant challenges for ap-
plying previous algorithms to LiDAR data in outdoor environments. Developing practical LiDAR-based algorithms to address these issues is currently a critical task.

To this end, we propose a novel NeRF-based LiDAR odometry and mapping method, dubbed NeRF-LOAM. It employs sparse octree-based voxels combined with neural implicit embeddings, decoded into a continuous signed distance function (SDF) by a neural implicit decoder. The embeddings, decoder, and poses are optimized simultaneously by minimizing the SDF errors. NeRF-LOAM targets the outdoor driving environments and separates the LiDAR points into ground and non-ground points, and a precise SDF for ground points can be obtained with the help of normals. Such an operation depresses Z-axis drift and smooths our dense 3D map. To tackle the incremental odometry and mapping under the unknown large-scale outdoor environment, a dynamic voxel embedding generation strategy without any pre-allocation or time-consuming loop is designed. Finally, we use key-scans to not only jointly refine the pose and the map but also relieve the catastrophic forgetting or pre-training process. Extensive experiments were conducted on three publicly available datasets. The experimental results demonstrate that our method attains state-of-the-art odometry and mapping performance in outdoor large-scale environments using LiDAR data.

To sum up, the contributions of our work are threefold:
1. To the best of our knowledge, our NeRF-LOAM is the first neural implicit odometry and mapping method for large-scale environments using LiDAR data.
2. We propose a novel neural SDF module combined with dynamic generation and key-scans refine strategy, which realizes a fast allocation of voxel embeddings in the octree and a high-fidelity 3D representation.
3. Based on the proposed online joint optimization, our method is pre-training free and generalizes well in different environments.

2. Related Work

Odometry and mapping in outdoor large-scale environments using LiDAR data has been investigated for decades. One of the primary methods is the iterative closest point (ICP) algorithm [3, 30], which directly aligns consecutive point clouds together and calculates the relative transformation between pairs of LiDAR scans. Tackling the sparsity of LiDAR data, Zhang and Singh [48] use point-to-edge and point-to-plane distance to optimize the ICP error and achieve accurate odometry estimates. However, these types of algorithms mainly focus on odometry estimation, while the reconstructed map is coarse. The successive research [3, 31, 2, 8] also explores the scene geometry to get more accurate odometry results without considering the quality of the reconstruction map. Meanwhile, learning-based methods on LiDAR odometry [15, 41, 26, 39, 6] attract much attention. These methods employ a network to learn features from points or projected 2D images. However, they often require large data for training and cannot generalize well to other environments.

To represent the 3D scene, there are many techniques such as surfels [28], occupancy grids [10], triangle meshes [19, 7], and polynomial representations [14]. Traditionally, Poisson surface reconstruction [23, 24] provides geometrically accurate reconstruction. Newcombe et al. [12] popularizes the concept of truncated signed distance function (TSDF) and volumetric integration methods to reconstruct triangle meshes [13, 37]. Behley and Stachniss [2] use surfels to realize the reconstruction of 3D range sensors. For learning-based reconstruction, they usually focus on the small objects [20] or reconstruct directly from the point clouds [43]. The dense reconstruction from 3D incremental LiDAR data still remains to explore.

Compared to the existing 3D representations, the success of neural implicit representation [1, 18, 32, 40, 50] for novel view synthesis attract great attention, and many research investigates the possibility to use this concept realizing simultaneous localization and mapping (SLAM) [42, 46, 27, 33, 51, 45]. These neural SLAM use multilayer perceptrons (MLPs) to represent the entire scene and achieve seminal results. Extensive related works have been done such as the training and inference speed [18, 17], sparse training view [47, 5] and scene composition [49, 44]. However, they are mainly designed to process the image [25, 32] or RGBD inputs [9, 4] and are employed indoors. Extending them to LiDAR-based outdoor environments is hard to achieve because of the model limitation of simple MLPs and the sparsity character of LiDAR data. Although [45, 50] adopt an octree-based sparse grid with voxel embeddings and can be applied in larger areas, the pre-allocated embeddings or time-consuming loop to search the voxels is not available in outdoor for both odometry and mapping.

Unlike the above-mentioned methods, we propose a novel neural implicit odometry and mapping method for incremental LiDAR inputs under large-scale environments to obtain both dense 3D representation and accurate poses. We adopt voxel embeddings with an MLP decoder to represent the local geometry instead of the entire scene, which generalizes well in most environments. We also design a dynamic voxel embedding generation strategy to reduce processing time significantly as well as a key-scans refine strategy to improve the reconstruction quality.

3. Our Neural SDF

Before delving into the details of our NeRF-LOAM network, we first introduce a novel neural SDF module shown in Fig. 2, which plays a crucial role in all of our processes, including optimizing the poses, maps, and networks.

To realize the neural representation of large-scale out-
Figure 2. The modified neural SDF. After the rays and points sampling, the voxel embeddings are fed to a network to query the neural SDF after ground separation.

doors incremental, the octree [21, 35, 34] structure is often adopted to recursively divide the scene into leaf nodes with basic scene units voxels. These axis-aligned voxels attach an $N_v$-dimension embedding at each vertex and share with neighbor voxels. The SDF values can be inferred from the embeddings through a neural network $F_\theta$. Different from existing methods [45, 50], we treat the environments differently when optimizing the SDF values, e.g., ground and non-ground, and propose a novel loss function to realize more suitable neural SDF for LiDAR data in outdoor large-scale environments.

Rays and points sampling. The first step in all of our processes is based on effective sampling. Instead of randomly selecting samples across the space or around the points, we first select rays that intersect with the currently allocated voxel and then select the points along the intersection part of the ray and voxels. Note that we set a hit number threshold of voxels $M_n$ to avoid the influence of the unseen surface. Since the LiDAR rays are transformed by the scan pose $T_i$, each ray contains the pose information of the scan. This sampling strategy allows us to optimize the pose and voxel embeddings simultaneously.

Neural SDF value. For most visual-based NeRFs [1, 51], the scalar value like weight or color are obtained by differentiable rendering along the ray. Since the SDF is a direct method to represent the scene, the rendering method is unsuitable for LiDAR data in outdoor environments [50]. The neural SDF filed $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ can be represented by Eq. (1): each sampled point can be regressed via the trilinear interpolation of voxel embeddings:

$$\Psi(p_s) = F_\theta(\text{TriInpo}(p_s, e_1, ..., e_n)),\ (1)$$

where $p_s = T_i p_f$ is the transformed sampled points by current scan pose $T_i$ from the original point $p_f$ in LiDAR coordinate, TriInpo($p_s, e_1, ..., e_n$) represents the trilinear interpolation of the sampled point $p_s$ surrounded by $n$ neighbor voxel embeddings, and $F_\theta$ is the neural implicit network with parameter $\theta$. Since all processes involved are differentiable, we can optimize the scan pose, voxel embeddings, and network parameters jointly through back projection. Because the voxel embeddings primarily store geometric information, our network does not require pre-training and can adjust online to different environments.

Training SDF pairs. The LiDAR sensors provide highly accurate range measurements, which allow us to compute the signed distance from the sampled points to the endpoints along the ray. This signed distance is often called the SDF value in many SLAM or mapping approaches [37, 13]. This approximation is generally acceptable for simple mapping or indoor SLAM tasks while leading to sub-optimal results when applied to outdoor SLAM as shown in Fig. 3. It illustrates the issue with the SDF approximation when used with a far LiDAR point. The blue line represents the true SDF value, while the red line is the SDF approximation. The difference between the two distances can be significant when the angle $\theta$ is close to $0^\circ$. This can decrease odometry quality due to the inaccurate SDF value. This problem is even more significant in the Z-axis, as there are fewer points in LiDAR scan to constrain Z-drift. While obtaining the normals of all LiDAR points can be challenging, the “smooth” ground allows access to the rectified SDF value.

Therefore, we propose to first separate LiDAR points into ground points $\mathcal{G}$ and non-ground points $\mathcal{G}^c$. The SDF field $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}$ can then be represented as:

$$\Phi(p_s) = \begin{cases} (p_s - p)n_p & \text{if } p \in \mathcal{G} \\ \|p_s - p\| & \text{else} \end{cases}, \ (2)$$

where $p_s$ is the sampled point and $p$ is the LiDAR point alone the ray, $n_p$ is the normal of point $p$.

Optimization. We train the network using the weighted sum of three different losses. The first free space loss forces the neural SDF of points between LiDAR and the positive truncation region $P_f$ to be truncation distance $T_r$:

$$L_f = \frac{1}{|P_f|} \sum_{i=0}^{|P_f|} (\Psi(p_i) - T_r)^2 \ (3)$$

The negative truncation region is beyond our consideration following the suggestion of [1] to avoid surface intersection ambiguities [40]. This loss plays an important role in removing dynamic objects. Secondly, we define an SDF loss of points within the truncation region $P_s$ to supervise the SDF estimates:

$$L_s = \frac{1}{|P_s|} \sum_{i=0}^{|P_s|} (\Psi(p_i) - \Phi(p_i))^2. \ (4)$$
4. NeRF-LOAM Framework

4.1. Overview

The architecture of our framework is illustrated in Fig. 4. Our method takes an incremental LiDAR stream as input and outputs a 3D reconstructed mesh with poses of each LiDAR scan through three modules: neural odometry, neural mapping, and mesh reconstruction. The first two parts run in parallel as frontend and backend, while the third runs separately. The input LiDAR scan is first processed by neural odometry to estimate the 6-DoF pose of the current scan in the world coordinate system. The pose is then used to transform the point cloud into the world coordinate system. Subsequently, any unvisited voxels will be added to the octree along with their corresponding voxel embeddings. To quickly locate the desired embeddings, we encode the 3D voxel coordinates into a unique scalar value, namely the Morton code [35]. Although utilizing the code, the pre-allocate embeddings [34, 45] or time-consuming one by one search in hash table [50] is not suitable for our task, especially when it needs to retrieve hundreds of thousands of embeddings from a hash table containing millions of entries.

Inspired by the concept of a look-up table, we devise an efficient and scalable method for generating voxel embeddings dynamically, as outlined in Alg. 1. The lookup table is extended with the maximum Morton code to store the access information of voxels. The unvisited voxels will be assigned initialized embeddings and added to the embedding table.
Algorithm 1: Dynamic Embeddings Generation.

Input: Look-up list \( L \); Incoming voxels IDs (i.e., Morton code) \( I_v \); Embedding list \( L_e \).
Output: Embedding list \( L_e \) with new embeddings; Updated look-up list \( L \).

1. \( m \leftarrow \max(I_v) \), maximum index.
2. \( l \leftarrow \text{len}(L) \), length of the look-up list.
3. \( s \leftarrow \text{len}(L_e) \), length of the embeddings list.
4. If \( m > l \) then
   5. \( L_e \leftarrow L_e[I_v] \), look the embeddings IDs.
   6. \( I_v \leftarrow \{ I_v[i] \mid I_v[i] = -1 \} \), unvisited voxels IDs.
   7. \( l_v \leftarrow \text{len}(I_v) \), length of unvisited quality.
   8. \( L_e' \leftarrow \{ e_v[i] \mid I_v[i] = -1 \} \), new embeddings.
   9. \( L_e \leftarrow L_e + L_e' \), final embedding list.
   10. \( L[I_v] \leftarrow [s + 1, \ldots, s + l + v] \), update the look-up list.

list while updating the look-up table by the current embedding number, eliminating time-consuming loop queries.

Joint optimization of the map and pose. Similar to neural odometry, we sample the rays and points to calculate the loss. Here we mainly optimize the voxel embeddings while fine-tuning the poses.

4.4. Mesh Reconstruction

Key-scans selection and refinement. We maintain a key-scan buffer to relieve the catastrophic forgetting of the first \( K \) scans as well as improve the mapping quality. A key scan is added to the buffer if the number of newly added voxels \( N_v \) exceeds a threshold of \( N_v \) or the distance between the current scan and the last key-scan \( d_f \) is sufficiently large. The map and poses are in the end refined with all the key-scans in the buffer. This simple strategy is effective, as demonstrated in the mapping results in Sec. 5.5. Additionally, to improve the efficiency of the refinement process, only rays or LiDAR points within a truncation distance \( d_f \) based on the point density are included.

Final mesh and poses. After the key-scans refine, the map and the poses are well-trained and ready to output final results. Our modified SDF is continuous, so we can theoretically infer SDF values at an arbitrary position. We query the SDF values with the same fixed size (i.e., voxel size), and the final mesh is obtained via marching cube [19].

5. Experiments

5.1. Experimental Setup

Datasets. We evaluate our method and compare it with state-of-the-art (SOTA) methods using three publicly available outdoor LiDAR datasets, including MaiCity [36], Newer College [29], and KITTI odometry [11] datasets. MaiCity [36] contains 64-beam noise-free synthetic LiDAR scans in urban environments, and the ground truth map is provided. Newer College [29] contains a hand-carried LiDAR sequence collected at Oxford University with motion distortion. To make it more challenging and the scans more distinctive, we take one out of every five. We compare our odometry and mapping results with provided ground truth trajectories and mesh maps by these two datasets. KITTI odometry [11] does not provide ground truth maps, so we present our odometry accuracy hereby qualitative mapping results.

Evaluation metric. We evaluate both the odometry and mapping performance of our method. For odometry accuracy, we present the root-mean-square error (RMSE) of absolute trajectory errors (ATEs) by \( SE(3) \) alignment. For mapping accuracy, we use the commonly used reconstruction metrics adopted in most reconstruction methods [22, 36, 50], i.e., accuracy, completion, Chamfer-L1 distance, and F-score, obtained by comparing the resulting mesh with ground truth.

Implemental details. The whole process shared network is an MLP consisting of 2 FC layers, and each layer has 256 hidden units. The length of our voxel embeddings is 16 with a voxel size 0.2 m. For sampling, we set the step size ratio to 0.2 for odometry and 0.5 for mapping and the truncation distance \( Tr = 0.3 \) m. To distinct the ground from the LiDAR points, we use the seminal work of [16]. More studies on our hyperparameter selection are presented in Sec. 5.5 and supplementary materials (see Sec. D).

5.2. Simultaneously Odometry & Mapping Results

The first experiment shows the simultaneous odometry and dense mapping results of our method compared with existing SOTA methods. For example, Poisson surface reconstruction SLAM method Puma [36], a TSDF fusion-based approach Vdbfusion [37], and an implicit neural network-based map representation SHINE-Mapping [50]. Since both Vdbfusion and SHINE-Mapping only focus on dense mapping, we combine them with the current SOTA odometry method KissICP [38]. For fair comparison, we also show the results of our methods using KissICP poses. The results of all baseline methods are produced using their official open-source code with the same voxel size.

Tab. 1 shows our odometry mapping results on the MaiCity[36] and Newer College[29] datasets. As can be seen, our mapping process combined with KissICP outperforms all baselines on the MaiCity dataset and has comparable quality in the Newer College dataset. The corresponding qualitative results are demonstrated in Fig. 5. In the case of the MaiCity dataset, KissICP produces false pose estimates in the initial scans, which will lead to entangled
mapping if there are no specific processes to remove these artifacts. Vdbfusion provides space carving to address this problem. However, it removes both the artifacts and important objects such as roads, trees, and cars. Shine-Mapping offers some improvement by removing certain artifacts. Our proposed method outperforms both of these techniques by effectively removing the majority of the artifacts and producing a smoother mapping result. Similar benefits can be observed in the Newer College dataset, where Vdbfusion removes the trajectory caused by a person holding a device, resulting in an incomplete map.

Compared to Puma, which involves both odometry and mapping processes, our approach also realizes both odometry and mapping using an implicit neural network and achieves superior performance in almost all metrics. In the MaiCity dataset, the slightly inaccurate trajectory of our method results in a larger distance compared to the completion distance, as also presented in Sec. 5.4. However, with a more precise trajectory in the Newer College dataset, our approach significantly outperforms Puma. These results are visually depicted in Fig. 6(d) and Fig. 6(e) for MaiCity, and Fig. 7(d) and Fig. 7(e) for Newer College. Although Puma appears more complete, the second row of the figures indicates that this comes at the expense of mapping accuracy. Also, on the MaiCity dataset, we can see the ground folds for Puma as it tries to reconstruct a watertight surface and thus is influenced by surrounding objects. As shown on the Newer College dataset, Puma cannot remove the dynamic objects and insufficient points on the wall hinder a complete reconstruction.

5.3. Mapping Quality

To eliminate the influence of pose estimation and thoroughly investigate the mapping ability of different methods, we employ ground truth poses to reconstruct the mesh map of the environments. We compare our approach with two pure mapping methods, Shine-Mapping [50] and Vdbfusion [37], and provide quantitative results in Tab. 2. As can be seen, our approach outperforms all baseline methods across almost all metrics when compared in terms of pure mapping ability. The superiority of our mapping approach is also evident in Fig. 6 and Fig. 7, where our reconstruction is the most complete, particularly in terms of the ground. The error maps enforce our claims by demonstrating the greater accuracy of our reconstruction. Note that in the Newer College dataset, we reconstruct every five scans, and the results indicate that our mapping process still performs well even with sparse and noisy observations.

5.4. Odometry Evaluation

As discussed, the quality of odometry largely influences the mapping quality. An accurate trajectory can directly improve the reconstruction result and avoid undesired artifacts. Here we present the results of our odometry compared with other non-learning-based and learning-based methods. As mapping methods like Shine-Mapping [50] and Vdbfusion [37] do not provide pose estimations, they are omitted from the comparison. For non-learning-based methods, we compare our odometry results with Puma [36], SuMA [2], and two registration algorithms based on ICP: point-to-point ICP [3] and generalized-ICP [31]. For learning-based methods, we adopt two SOTA algorithms with code available: DeLORA [26] and PWC-LONet [39]. For other code-unavailable learning-based methods like LO-Net [15] and DeepPCO [41], we report their quantitative results from their papers in our supplementary materials (see Sec. C) along with the above-mentioned methods.

We present the RMSE results in Tab. 3. Our method achieves comparable results to other methods on the synthetic MaiCity dataset and KITTI09 datasets while achieving the best performance on the Newer College. Notably, our method does not require any pre-training and exhibits strong generalization ability across different datasets, while pre-trained methods such as DeLORA and PWC-LONet,
which are pre-trained on the KITTI dataset, exhibit worse performance on other datasets. Although PWC-LONet still obtains acceptable results on the MaiCity dataset, it almost fails on the Newer College dataset. More results on KITTI can be found in the supplementary materials (see Sec. C).

5.5. Ablation Study

Ground separation. We compare the performance of our method with and without ground separation and show the odometry and mapping accuracy in Tab. 4. For odometry accuracy, we see that RMSE error declines with ground separation for the MaiCity dataset and for Newer College, the approach even failed without ground separation. Moreover, when checking pose error in each axis (supplementary materials Sec. D), the trajectory with ground separation is consistent in the z-axis, while without ground separation, it diverges fast. For mapping accuracy, all mapping metrics indicate that our method achieves significantly better mapping results with ground separation. We can also see a clear improvement visually in Fig. 8. With ground separation, the “ripples effect” is suppressed and the holes are disappeared.

Key-scan refine strategy. We further analyze the effectiveness of our key-scan refine strategy and show the result in Tab. 4. The numerical results show improvement with key-scan refinement, and the visual improvement is even more significant, as shown Fig. 9. The key-scan refinement produces smoother and more complete results, as evidenced by the improved maps of roads, walls, and vehicles.

Voxel size. We analyze the mapping quality, memory consumption, and processing time v.s. the voxel size shown
Table 2. The mapping results of the reconstruction quality on the MaiCity [36] and Newer College [29] dataset. The voxel size is 20 cm and F-score in % with a 10 cm error threshold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pose</th>
<th>MaiCity</th>
<th>Newer College</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHINE [50]</td>
<td></td>
<td>4.17</td>
<td>5.30</td>
</tr>
<tr>
<td>Vdbfusion [37]</td>
<td>GT pose</td>
<td>4.12</td>
<td>8.01</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>3.15</td>
<td>4.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.32</td>
<td>14.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>MaiCity</th>
<th>Newer College</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.17</td>
<td>5.30</td>
</tr>
<tr>
<td></td>
<td>8.32</td>
<td>14.36</td>
</tr>
</tbody>
</table>

Table 3. RMSE results of odometry. Mai for MaiCity [36], NC for Newer College [29], KT for KITTI [11], "-" for failed.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ground KF-ref.</th>
<th>RMSE↓</th>
<th>Acc.↓</th>
<th>Comp.↓</th>
<th>C-l1.↓</th>
<th>F↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaiCity</td>
<td></td>
<td>0.20</td>
<td>6.15</td>
<td>69.64</td>
<td>37.90</td>
<td>49.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.20</td>
<td>6.13</td>
<td>70.48</td>
<td>38.30</td>
<td>48.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.17</td>
<td>5.93</td>
<td>11.49</td>
<td>8.71</td>
<td>76.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.17</td>
<td>5.69</td>
<td>11.23</td>
<td>8.46</td>
<td>77.26</td>
</tr>
<tr>
<td>Newer College</td>
<td></td>
<td>0.20</td>
<td>6.15</td>
<td>69.64</td>
<td>37.90</td>
<td>49.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.20</td>
<td>6.13</td>
<td>70.48</td>
<td>38.30</td>
<td>48.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.17</td>
<td>5.93</td>
<td>11.49</td>
<td>8.71</td>
<td>76.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.17</td>
<td>5.69</td>
<td>11.23</td>
<td>8.46</td>
<td>77.26</td>
</tr>
</tbody>
</table>

Table 4. Ablation study of our designs on MaiCity [36], Newer College [29]. "-" stands for failed.

6. Conclusion

In this paper, we presented a novel approach for simultaneous odometry and mapping using neural implicit representation with 3D LiDAR data. The devised NeRF-LOAM network tackles incremental LiDAR inputs in outdoor large-scale environments. It uses voxel embeddings to record the geometrical structure and avoids any pre-training, thus generalizing well in different situations. We further conceive a dynamic embedding generation, which realizes fast query and allocation to support outdoor large-scale applications. Experiments conducted on simulated and real-world datasets showed that our approach reconstructs higher-quality 3D mesh maps compared to other learning-based or non-learning-based methods. Our method estimates at the same time an accurate pose and generalizes well without any offline pre-training.

Limitation and future work. Our NeRF-LOAM cannot currently operate in real-time with our unoptimized Python implementation. The primary bottleneck is the intersection query between the ray and the map. For future work, we can facilitate the runtime by using sliding windows or local searching based on the estimated odometry pose and optimize the code in C++. Additionally, we plan to combine our work with loop closures to handle drift in long-term tracking and mapping, ultimately achieving a full SLAM system.

Acknowledgements. This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant 62273229, and in part by ByteDance project under Grant CT20220323003446, and also in part by Shanghai Slamtec Co., Ltd., with whom completed in cooperation.
References


