Hierarchical Pose Estimation and Mapping with Multi-scale Neural Feature Fields

1st Evgenii Kruzhkov
Autonomous Intelligent Systems
Computer Science Institute VI, University of Bonn
Bonn, Germany, ekruzhkov@ais.uni-bonn.de

2nd Alena Savinykh
Autonomous Intelligent Systems
Computer Science Institute VI, University of Bonn
Bonn, Germany, asavinykh@ais.uni-bonn.de

3rd Sven Behnke
Autonomous Intelligent Systems, Computer Science Institute VI – Intelligent Systems and Robotics,
Center for Robotics, and Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn
Bonn, Germany, behnke@cs.uni-bonn.de

Abstract—Robotic applications require a comprehensive understanding of the scene. In recent years, neural fields-based approaches that parameterize the entire environment have become popular. These approaches are promising due to their continuous nature and their ability to learn scene priors. However, the use of neural fields in robotics becomes challenging when dealing with unknown sensor poses and sequential measurements. This paper focuses on the problem of sensor pose estimation for large-scale neural implicit SLAM. We investigate implicit mapping from a probabilistic perspective and propose hierarchical pose estimation with a corresponding neural network architecture. Our method is well-suited for large-scale implicit map representations. The proposed approach operates on consecutive outdoor LiDAR scans and achieves accurate pose estimation, while maintaining stable mapping quality for both short and long trajectories. We built our method on a structured and sparse implicit representation suitable for large-scale reconstruction and evaluated it using the KITTI and MaiCity datasets. Our approach outperforms the baseline in terms of mapping with unknown poses and achieves state-of-the-art localization accuracy.

Index Terms—Hierarchical Pose Optimization, Structured Neural Fields, Large-Scale Implicit Mapping, Implicit SLAM, Sequential Data Training
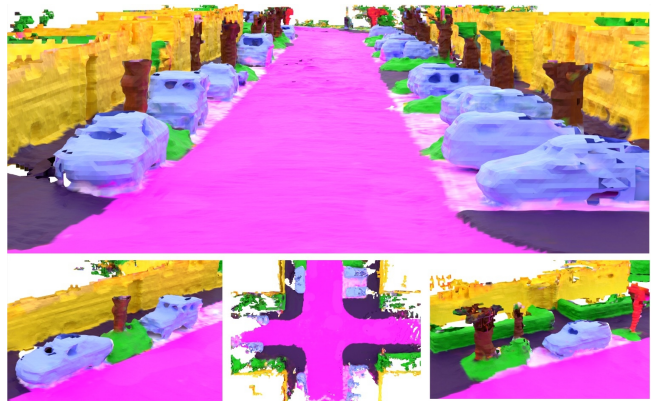
Fig. 1. Demonstration of the flexibility of the proposed implicit mapping extended to semantic domain. We create the map using sequential LiDAR measurements and corresponding semantic labels from the SemanticKITTI dataset [12]. Our approach effectively learns a 3D semantic representation from the data, demonstrating its generalization abilities. We employ the Marching Cubes algorithm [13] to visualize the learned information.

I. INTRODUCTION

Pose estimation and mapping are essential tasks that have a significant impact on mobile robotics, self-driving cars, virtual reality, and computer vision. These two tasks are closely related, as the quality of one influences the quality of the other. The combined task is commonly known as simultaneous localization and mapping (SLAM). In recent years, researchers have explored novel approaches to high-quality and efficient mapping with accurate pose estimation [1]–[4].

Although significant progress has been made in SLAM, it remains necessary to develop novel methods to meet the increasing demands for quality and generalization ability. For instance, sparse methods [5] produce maps that have limited applicability (mainly further localization) because they do not represent dense surfaces or volumes. On the other hand, dense [6] and semi-dense [7] maps can capture a greater amount of information and finer details allowing numerous downstream tasks. However, they require more computational resources and still have limited expression capability because of their discrete nature.

Today, numerous robotic applications [8] demand a more advanced understanding of the environment, which calls for the development of novel comprehensive maps. To address this, some works [9]–[11] introduce parameterized elements to the scenes (doors, planes, small items, etc.). However, these approaches can parameterize only a portion of the scene's objects, which can be sufficient for some scenarios but does not provide a general solution.

Recently, Neural Radiance Fields (NeRF) [14]–[16] have gained attention in the scientific community. Although most of these approaches rely on ground truth poses that are unavailable for real-time robotics applications, they still possess useful properties. For instance, neural fields can parameterize the entire scene from plain geometry to fine-grained details [17]–[20]. Moreover, this representation can easily be extended to additional levels of scene understanding [21]. Figure 1 illustrates the semantic mapping achieved by the proposed

approach that is augmented to represent semantic information alongside 3D geometry (Section III-B).

In this paper, we advance the current state of research on neural implicit representations for robotic applications. Specifically, we focus on implicit representations of large-scale environments and simultaneous pose estimation inside neural fields. We provide a probabilistic interpretation of implicit mapping and introduce coarse-to-fine pose optimization based on sequential LiDAR data for large-scale neural field maps, enabling operations in street-sized environments.

In summary, we present in this work:

- structured sparse implicit mapping and pose estimation with probabilistic interpretation,
- hierarchical pose optimization based on the octree map structure, and
- evaluation of our proposed approach relative to state-of-the-art methods.

II. RELATED WORKS

CodeSLAM [22] is one of the pioneering works in SLAM that proposes a compact auto-encoder based implicit environment representation. However, in recent years, NeRF-based architectures [14] have gained popularity in the field of implicit mapping, leading to the development of novel SLAM algorithms. Deng *et al.* [23] demonstrated that providing additional depth information improves the training convergence of such architectures. For instance, iMap [24] is one of the first approaches that uses neural radiance fields and an RGB-D sensor for SLAM, but it can only operate in small environments.

Several studies have advanced neural fields to encompass larger scales. For instance, KiloNeRF [25] proposes the use of tiny voxel-assigned neural networks to accelerate learning and inference. Similarly, MeSLAM [26] utilizes coupled neural field blocks to achieve scalability. Müller *et al.* [18] introduce multiresolution hash encoding, which considerably improved training speed and network generalization capabilities, but it is not easily applicable to unbounded scenes. In contrast, Nice-SLAM [27] implements a grid structure that enables mid-size environment mapping, but it is inefficient for open space scenes.

Studies including [24], [26], [27] sample points along the rays and aggregate predictions at these sampled points to derive a loss for training neural fields. However, this approach can be computationally inefficient for open-space environments. In contrast, we consider a probabilistic interpretation of the mapping task (Sec. III-A) and derive direct optimization to avoid computationally expensive volume rendering. Our method utilizes octree-based feature grids [28], making it suitable for open-space environments.

Active research on gradient-based pose optimization is conducted in works [29], [30]. BARF [19] proposes coarse-to-fine pose optimization, which facilitates neural field training with imprecise camera poses. However, this method assumes the instant availability of all data and is designed for specific positional encoding [31]. Pure localization in neural fields is

also studied, as seen in [32], [33], which focus on Monte Carlo localization [34] in implicit maps.

Large-scale neural rendering reconstruction is addressed in [17], [35], but these approaches are not easily adaptable to SLAM. Moreover, approaches based on [14] suffer from artifacts in the reconstructed 3D geometry. To address this issue, Azinović et al. [36] propose to directly predict a truncated signed distance function (TSDF). Zhong et al. [37] combine an efficient octree structure with the prediction of TSDF to achieve a precise large-scale 3D reconstruction.

The work most similar to ours is SHINE-Mapping [37], which introduces a mapping-with-known-poses method. We use it as a baseline to estimate the mapping quality of our approach. In contrast to related works, our method enables large-scale implicit mapping and hierarchical pose estimation in neural fields. It handles sequential 3D LiDAR data and does not require ground-truth poses, making it suitable for practical robotic applications.

III. METHOD

This section describes the components of our method. We first introduce a probabilistic interpretation of the proposed approach. Then we describe the proposed neural network architecture in depth. Finally, we present the details of our pose estimation in a hierarchical, coarse-to-fine manner.

Fig. 2 illustrates the core components of our method. The proposed network architecture is trained to represent the map with varying levels of detail using 3D LiDAR data as input.

Based on the previous pose estimations, we optimize the current sensor pose through gradient flow defined by loss between current LiDAR measurements and the learned implicit map. As a result, our approach performs simultaneous pose estimation and implicit mapping based on sequential LiDAR data.

A. Mapping

Information about the environment is obtained from measurements of noisy sensors. We consider the environment to be a probability density function (PDF) of the measured noisy values. We employ a neural field-based model to approximate the PDF of the environment. To learn the model, we minimize the KL divergence between the approximated and real PDFs, denoted $q(x, w)$ and $p(x)$, respectively. Minimizing D_{KL} is equivalent to minimizing cross-entropy [38] and maximizing the log-likelihood of $q(x, w)$:

$$\min_{w} D_{KL}\left(\frac{p(x)}{q(x, w)}\right) \sim \max_{w} \frac{1}{N} \sum_{i=1}^{N} log(q(x_i, w)), \quad (1)$$

where w represents the weights of the neural field model, x is a 3D spatial coordinate and N is the number of sampled points.

We assume that the sensor measurements $y(x_i)$ have normally distributed noise with mean μ and variance σ^2. Then the measured environment has normally distributed PDF:

$$q(x_i, w|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y(x_i)-\mu)^2}{2\sigma^2}}. \quad (2)$$
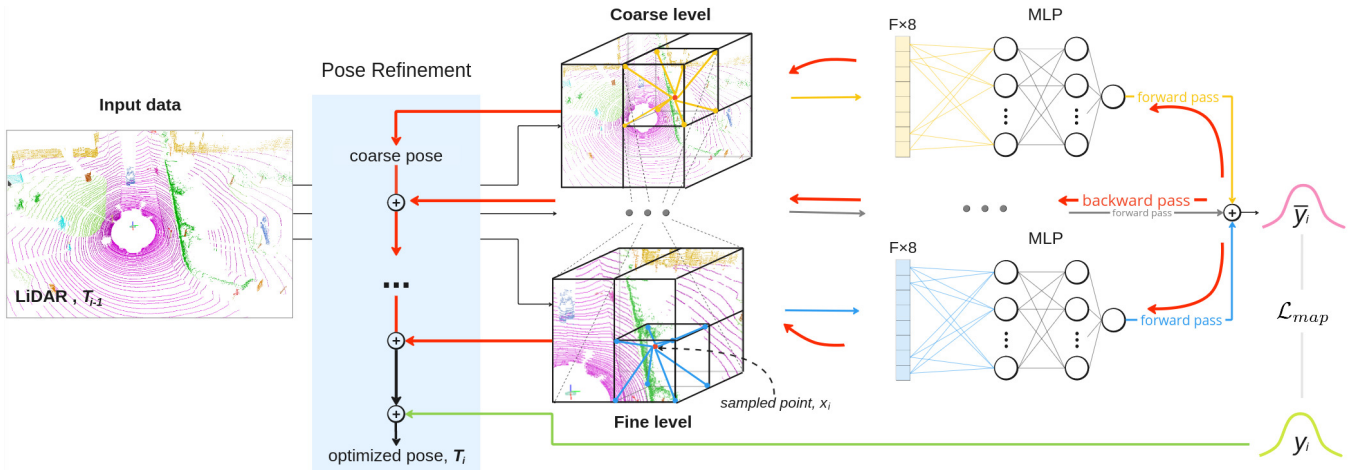
Fig. 2. Overview of the proposed approach. Our method utilizes neural fields consisting of learnable F-dimensional features stored in the corners of an octree-based structure. Each trainable octree level is associated with tiny MLP networks. During the forward pass, the LiDAR measurements are transformed to world coordinates using an initial transformation T_{i-1}. The features of the voxel corners are then weighted based on the relative position of the sampled point x_i (red) from the measurements. These weighted features are concatenated and fed to the corresponding MLP. The predictions of all layers are accumulated to generate the final occupancy probability $\bar{y}_i$ for the sampled points. During the backward pass, the gradient values are backpropagated to optimize the transformation T_{i-1} toward T_i in two ways: directly through measurements y_i and hierarchically though $\bar{y}_i$ (Sec. III-C).

Substituting Eq. 2 into Eq. 1, the likelihood maximization becomes a root mean square error minimization (RMSE) problem:

$$\mathcal{L}_{map} = \frac{1}{N} \sum_{i=1}^{N} (y(x_i) - \bar{y}(x_i))^2, \quad (3)$$

where $\bar{y}(x_i)$ represents the prediction of the model at coordinate x_i and $y(x_i)$ represents the mean μ of the measurements. We choose the signed distance function (SDF) as the environment representation. We acquire environment observations $y(x_i)$ by sampling points around LiDAR point clouds and approximating $y(x_i)$ as the distance to the closest point in the LiDAR measurements. Both $y(x_i)$ and $\bar{y}(x_i)$ are direct values of x_i obtained without volume rendering.

B. Network Architecture

We employ learnable F-dimensional features that are fed to MLPs as our neural field representation. These features are stored in corners of an octree structure (Fig. 2). Although the final reconstruction is a dense SDF, the octree sparsely stores learnable features only for the voxels where LiDAR measurements were observed. We use different octree levels and explicitly train them to learn coarse, mid-level and fine representations of the environment.

First, each layer is independent and has its own MLP neural network. The level's learnable features, combined with spatial coordinates of a queried point (Sec. III-C), serve as input to the network that infers the occupancy probability for the point. The combined features are concatenated, which, based on our observations, yields better prediction quality compared to the mean and sum input reductions.

Second, each MLP is explicitly guided to represent only its level details during mapping. To achieve this, we deactivate all layers except the coarsest one for the first iterations of

Eq. 3 optimization. Then, we activate finer levels and their corresponding MLPs one by one, with an equal iteration interval. The predictions from the newly activated levels are added to the coarser ones, guiding the learning of the finer details. We follow this technique for each mapping step.

Finally, we sum the predictions from all levels to obtain both low- and high-frequency details on the final map. Fig. 3 demonstrates the difference between the coarse-level prediction and the final prediction. The coarse level represents rough geometry, while the finer levels capture high-frequency details, resulting in improved map quality when combined together. With the proposed technique, predictions of the finer levels are guided to complement the coarser ones without redundantly learning the geometry representations of the previous levels.

We observed that finer octree levels are resistant to catastrophic forgetting if the input measurements are dense and the close measurements have sufficient density. Moreover, the close measurements overlap with the environment which is already mapped from the past observations. Therefore, we address catastrophic forgetting differently for close and far LiDAR measurements. For the far points, we tackle the problem regularizing the learnable levels' features similar to Zhong et al. [37]:

$$\mathcal{L}_{k,reg} = \gamma_k \sum \Omega(\mathbf{w}^t - \mathbf{w}^{t-1}), \quad (4)$$

where $\mathbf{w}^t$ and $\mathbf{w}^{t-1}$ represent the current and previously converged levels' features, respectively, Ω are importance weights proposed by [37], and γ_k is a level-k regularization weight introduced in our work. We propose higher weights γ_k for coarse levels to prevent forgetting of the whole geometry and smaller weights for the finer levels to allow updating of details.

For close measurements, we regularize outer octree voxels by sparsely reconstructing the points sampled at the fine

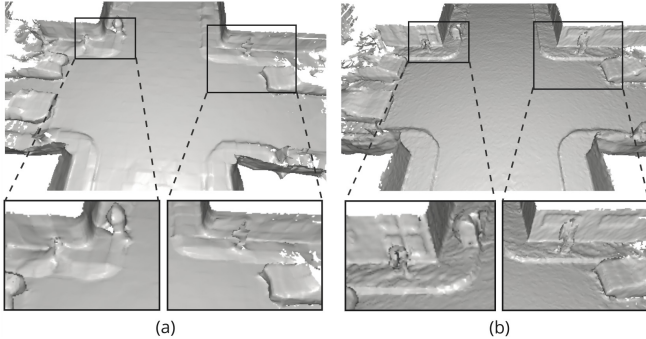| KITTI Method	Seq. 00 (3711 m)		Seq. 05 (2202 m)		Seq. 03 (567 m)		Seq. 07 (693 m)	
	ATE [m]	ATE [%]	ATE [m]	ATE [%]	ATE [m]	ATE [%]	ATE [m]	ATE [%]
ICP	37.55	1.04	11.80	0.53	5.06	0.89	7.68	1.11
KISS-ICP	6.28	0.17	**1.94**	**0.09**	3.35	0.59	**0.758**	**0.11**
Ours	**4.87**	**0.13**	4.07	0.18	**1.47**	**0.26**	0.87	0.13

Fig. 3. The coarse and final representations learned by our proposed approach, reconstructed using the Marching Cubes algorithm. (a) shows the representation learned solely by the coarse level MLP, while (b) displays the final representation that includes high-frequency features from the finer levels.

level and then adding them to the measurements as replay data, thus employing the hierarchical structure of our implicit representation. Our final loss for joint training of the levels' features and MLPs becomes:

$$\mathcal{L} = \mathcal{L}_{map} + \sum_k \mathcal{L}_{k,reg}. \qquad (5)$$

We observe that due to the sparse nature of the octree architecture, we can extend it to additional domains without drastically increasing the consumed memory. To generate Fig. 1, we extended the dimensions of the input with encoded semantic labels, increased the size of the features stored in the corners of the octree, and optimized the model to reconstruct encoded semantic labels together with the geometry using the same Eq. 3.

C. Pose Estimation

We perform simultaneous localization by optimizing the sensor pose in the neural fields. Having approximation $q(x, w)$ of the environment and current measurements $y(x_i)$, we optimize the sensor pose T_i which is used to project measurements from the sensor's local coordinates to the global coordinates. According to Eq. 3, it can be done through both $y(x_i)$ and $\bar{y}(x_i)$ but the path to T_i through the measurements $y(x_i)$ is more straightforward than through model prediction $\bar{y}(x_i)$. The meaning of such optimization is to find the position of the sensor where its observations correspond to the mapped environment $q(x, w)$ with the smallest RMSE. We use the previously estimated pose T_{i-1} as the initial one for each new optimization step.

During the forward pass, we weigh the features of each octree level according to the relative position of the sampled point x within its voxel before feeding them to the network:

$$z_j = h_j \cdot \prod (1 - x_s - \lfloor x_s \rfloor), \qquad (6)$$

where z_j represents the weighted j-corner features, h_j are the corner's F-dimensional features, and x_s is the sampled point position scaled to the level's grid resolution.

The weighted features of the same voxel are concatenated and fed into the corresponding MLP model, as shown in Fig. 2. With this structure, we propagate the gradient to the sensor pose directly through $y(x_i)$.

We additionally employ the proposed hierarchical architecture (Sec. III-B) and perform coarse-to-fine pose estimation. We first perform a coarse pose estimation through the coarse level and then gradually activate the finer levels to refine the pose through more detailed maps. According to our observations, pose optimization is more stable when performed in such a coarse-to-fine manner. Our final pose estimation consists of the direct (Fig. 2: green line) and coarse-to-fine optimizations (Fig. 2: red lines) performed concurrently.

IV. EXPERIMENTS

We evaluated the accuracy of the proposed hierarchical pose estimation in the KITTI dataset [12], which contains outdoor LiDAR sequences of different lengths. To assess the localization accuracy, we use the absolute trajectory error (ATE). We also measure the influence of pose estimation on mapping quality using the synthetic MaiCity dataset [39].

Throughout the experiments, we used unified parameters for localization and mapping. We set the feature dimension $F = 3$ and use 3 octree levels with learnable features. All tiny MLPs have 2 hidden layers with 32 neurons each.

Our approach takes 50 iterations for mapping. For pose estimation, we set 80 iteration: 20 iterations for coarse estimation, 20 iterations with the middle level activated, and the remaining 40 iterations with all levels activated.

A. Pose Estimation

The aim of this experiment is to verify that the proposed approach can effectively perform continuous localization throughout the whole sequence with decent and stable accuracy. We compare its performance with point-to-plane ICP [40] and state-of-the-art KISS-ICP [1]. The parameters of these methods are fixed for all sequences.

The results are presented in Tab. I. The proposed approach is superior to ICP and comparable to the state-of-the-art

TABLE II
MAPPING QUALITY RESUTLS ON THE MAICITY DATASET

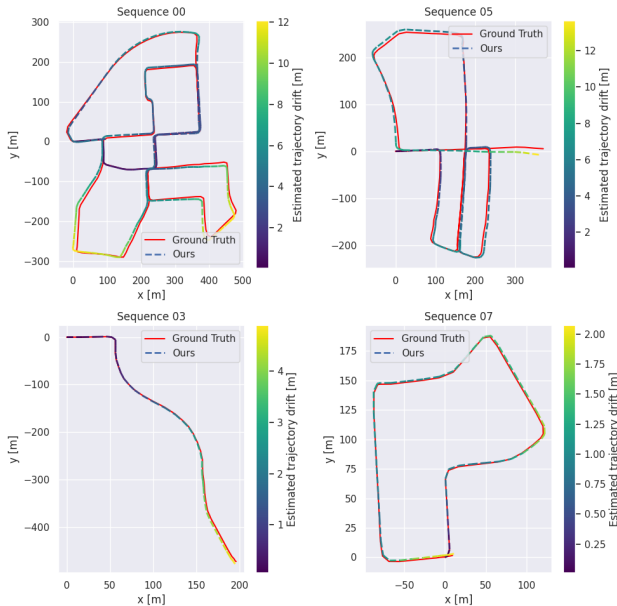| MaiCity	Seq. 00 (700 m)				Seq. 01 (100 m)			
Method	Acc.	Comp.	C-L1[1]	C-L2[2]	Acc.	Comp.	C-L1[1]	C-L2[2]
SHINE (GT)	**0.28**	**0.12**	**0.20**	**0.30**	0.26	0.07	0.17	0.26
SHINE (ICP)	5.92	0.46	3.19	5.90	0.35	0.20	0.28	0.36
Ours	1.0	0.29	0.65	1.2	**0.11**	**0.07**	**0.09**	**0.17**

[1] Chamfer-L1 distance.
[2] Chamfer-L2 distance.

Fig. 4. Estimated trajectories of the proposed approach on the KITTI dataset. The colorbar visualizes the distance between the poses of the estimated and ground truth trajectories.
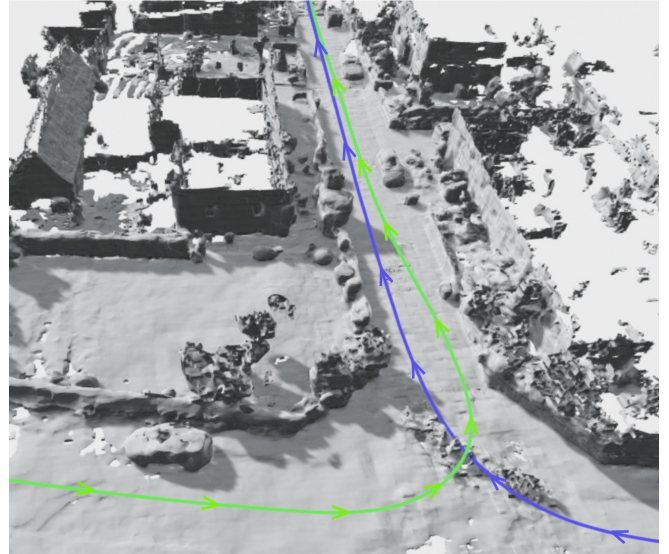

Fig. 5. Revisiting of previously mapped area. The map is initialized during the first visit of the region (green). During the second traversal (blue), the path successfully converges to the path taken during the first visit. Both traversals have the same motion direction.

KISS-ICP. Based on the experimental results, our method can provide consistent localization results for both short (Seq. 03 and Seq. 07) and long sequences (Seq. 00 and Seq. 05).

Fig. 4 shows our estimated trajectories and ground truth data for each sequence. We observe that the estimated trajectory usually follows the ground truth precisely. Further analysis reveals that the majority of residuals come from inaccuracy in orientation optimization around the pitch angle.

The proposed approach possesses good capabilities to handle places previously visited. For instance, trajectories in Seq. 00 and Seq. 05 shown in Fig. 4 have higher accuracy (darker color) when the place is visited more than once. Thus, our method can successfully re-localize itself and decrease localization error when operating in re-visited environments. We illustrate this in Fig. 5 which shows a street traversed twice in the same direction. The corresponding trajectories are colored in different colors. The blue trajectory corresponds to the second traversal and it converges to the first visit that is shown in green.

B. Mapping

This experiment verifies the quality of the map in short and long sequences of the MaiCity dataset [39]. We compare the proposed approach against SHINE-Mapping [37], which is similar to our mapping. The comparison results in terms of accuracy (Acc.), completeness (Comp.), and Chamfer distances (C-L1, C-L2) are presented in Tab. II.

We feed the LiDAR data to algorithms in a sequential manner since future measurements are unknown in the case of real-time operation. SHINE-Mapping is a pure mapping approach; thus it uses ground truth poses to infer a map. For comparison, we use SHINE-Mapping with ground truth poses (the best scenario) and poses estimated by the ICP method.

According to Tab. II, for the short Seq. 01 our approach achieves more accurate mapping results compared to both variants of SHINE-Mapping. This occurs because SHINE-Mapping is a reconstruction approach that demonstrates the best results when all measurements are available at the same time, while our approach is initially designed to handle sequential data at each time step and benefits from hierarchical optimizations. As expected, the reconstruction quality of our method is slightly worse than SHINE-Mapping with ground

truth poses for the long Seq. 00, but the accuracy values of both methods are still close. The increased mapping error in the longer sequence is caused by an accumulated localization error that slowly deforms the map. However, the mapping accuracy of our method has similar values in both short and long sequences. All these highlights the stability of the proposed pose estimation throughout sequences with various length.

Based on the experimental results, both SHINE-Mapping and our method are capable of providing an accurate map. However, SHINE-mapping quality depends on the localization accuracy estimated by third-party algorithms, while the proposed approach has stable built-in localization.

C. Real-Time Operation Discussion

The efficiency of our approach depends primarily on the size of the environment, the number of optimization steps per mapping and tracking, and the amount of data in the current measurement. We estimate the time efficiency on a laptop with an embedded NVIDIA GeForce RTX 3070 Ti and 16 GB of RAM memory, noting that modern autonomous robots can have even more powerful computational resources. For large KITTI sequences (approx. 4000 m), we use the fine level is 13th level of the octree with a resolution of 0.25 meters. In such conditions, our approach can operate at a frequency of 3 fps. For smaller environments (approx. 50 m), we achieve the frequency of 5 fps by using the 10th level as the finest level, with a resolution of 0.05 m.

Performance can be further increased by adjusting the parameters for each specific case. Efficient operation was achieved thanks to the sparse octree structure, depth-guided training, and the avoidance of costly volume rendering.

V. CONCLUSION

In this work, we present a pipeline for simultaneous mapping and pose estimation in structured implicit representations. The proposed neural network architecture, which includes the training of features in the octree corners on different levels and following hierarchical pose optimization, is the core for achieving a large-scale yet detailed map without ground-truth poses. Our approach works with real-world LiDAR data stream data and operates at 3-5 frames per second.

We perform an evaluation of the proposed approach on public datasets. It shows that our method is capable of simultaneous localization while preserving high mapping quality for large-scale environments. The localization accuracy verified in the KITTI dataset is on par with the state-of-the-art performance demonstrated by KISS-ICP.

To validate mapping quality, we compare our method with a state-of-the-art reconstruction baseline approach on the MaiCity dataset. Experimental results show that our mapping is superior when sequential LiDAR data is utilized.

ACKNOWLEDGMENT

REFERENCES

[1] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[2] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems (RSS)*, vol. 2, 2014, pp. 1–9.

[3] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D Lidar-based online mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5000–5007.

[4] J. Quenzel and S. Behnke, "Real-time multi-adaptive-resolution-surfel 6D LiDAR odometry using continuous-time trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5499–5506.

[5] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Le Besnerais, "OV^{2}SLAM: A fully online and versatile visual SLAM for real-time applications," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 1399–1406, 2021.

[6] K. M. Jatavallabhula, G. Iyer, and L. Paull, "∇SLAM: Dense SLAM meets automatic differentiation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2130–2137.

[7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[8] K. Yadav, J. Krantz, R. Ramrakhya, S. K. Ramakrishnan, J. Yang, A. Wang, J. Turner, A. Gokaslan, V.-P. Berges, R. Mootaghi, O. Maksymets, A. X. Chang, M. Savva, A. Clegg, D. S. Chaplot, and D. Batra, "Habitat challenge 2023," https://aihabitat.org/challenge/2023/, 2023.

[9] M. Nieuwenhuisen, J. Stückler, and S. Behnke, "Improving indoor navigation of autonomous robots by an explicit representation of doors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4895–4901.

[10] E. Sucar, K. Wada, and A. Davison, "NodeSLAM: Neural object descriptors for multi-view shape reconstruction," in *International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 949–958.

[11] J. Wang, M. Rünz, and L. Agapito, "DSP-SLAM: Object oriented SLAM with deep shape priors," in *International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1362–1371.

[12] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI dataset."

[13] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Seminal Graphics: Pioneering Efforts that Shaped the Field*. ACM, 1998, pp. 347–353.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[15] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4606–4613, 2022.

[16] M. Kurenkov, A. Potapov, A. Savinykh, E. Yudin, E. Kruzhkov, P. Karpyshev, and D. Tsetserukou, "NFOMP: Neural field for optimal motion planner of differential drive robots with nonholonomic constraints," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 10 991–10 998, 2022.

[17] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable large scene neural view synthesis," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8248–8258.

[18] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[19] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "BARF: Bundle-adjusting neural radiance fields," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 5741–5751.

[20] R. A. Rosu and S. Behnke, "PermutoSDF: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 8466–8475.

[21] S. Zhi, E. Sucar, A. Mouton, I. Haughton, T. Laidlow, and A. J. Davison, "iLabel: Revealing objects in neural fields," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 832–839, 2023.

[22] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "CodeSLAM – Learning a compact, optimisable representation for dense visual slam," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2560–2568.

[23] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 882–12 891.

[24] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit mapping and positioning in real-time," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 6229–6238.

[25] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 14 335–14 345.

[26] E. Kruzhkov, A. Savinykh, P. Karpyshev, M. Kurenkov, E. Yudin, A. Potapov, and D. Tsetserukou, "MeSLAM: Memory efficient slam based on neural fields," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 430–435.

[27] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural implicit scalable encoding for SLAM," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 786–12 796.

[28] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler, "Neural geometric level of detail: Real-time rendering with implicit 3D shapes," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 358–11 367.

[29] Z. Teed and J. Deng, "Tangent space backpropagation for 3D transformation groups," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 338–10 347.

[30] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF–: Neural radiance fields without known camera parameters," *arXiv preprint arXiv:2102.07064*, 2021.

[31] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Ragha-van, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *NeurIPS*, 2020.

[32] L. Wiesmann, T. Guadagnino, I. Vizzo, N. Zimmerman, Y. Pan, H. Kuang, J. Behley, and C. Stachniss, "LocNDF: Neural distance field mapping for robot localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 8, pp. 4999–5006, 2023.

[33] H. Kuang, X. Chen, T. Guadagnino, N. Zimmerman, J. Behley, and C. Stachniss, "IR-MCL: Implicit representation-based online global localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 3, pp. 1627–1634, 2023.

[34] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots," in *16th National Conference on Artificial Intelligence (AAAI) and 11th Conference on Innovative Applications of Artificial Intelligence (IAAI)*, vol. 1999, 1999, pp. 343–349.

[35] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the wild: Neural radiance fields for unconstrained photo collections," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7210–7219.

[36] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6290–6301.

[37] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "SHINE-Mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8371–8377.

[38] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, vol. 31, 2018, pp. 8792–8802.

[39] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss, "Poisson surface reconstruction for LiDAR odometry and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5624–5630.

[40] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.