Improving LONER with Semantic Segmentation

***Abstract*—LONER [1] is the first real-time LiDAR SLAM algorithm that uses NeRF as scene representation. It uses LiDAR data to train a NeRF representation in real-time with its novel loss function. In this project, we aim to build upon the LONER system by adding another feature to deal with movable objects in the Lidar scans. We integrate SalsaNext [2], a state-of-the-art real-time Lidar semantic segmentation model for movable object detection, and modify the system to ignore the corresponding Lidar rays during the training of LONER. We evaluate our new feature on the open-source dataset FusionPortable [3] and show that the trajectory estimation is improved upon the original LONER in many aspects, especially in scenes with many moving cars.**

I. Introduction

Neural Radiance Fields (NeRFs) [4] related methods are promising approaches for implicitly representing maps in robotics applications. NeRF use a Multi-Layer Perceptron (MLP) to estimate a function that maps a point in space to a volume density and color, which has advantages over traditional map representations like point clouds and occupancy grids. One significant advantage is that NeRF allows for the continuous querying of any point in the scene without discretization. NeRFs also offer the ability to produce realistic renders and create a mesh representation or novel views of the scene. Therefore, a SLAM algorithm that employs NeRFs for map representation would be highly beneficial for robotics applications.

Several works employ NeRFs for map representation in SLAM tasks. iMAP [5] represents the underlying scene with a single multi-layer perceptron. NICE-SLAM [6] tried to scale the method to larger maps by representing scenes as trainable hierarchical feature grids. NeRF-SLAM [7] uses DROID-SLAM [8] for tracking and a probabilistic NeRF [9] for scene representations.

LONER [1] proposes the first real-time LiDAR NeRF SLAM method. It showed competitive on-site trajectory estimation by using a novel loss function and a multi-resolution feature grid as Neural Radiance Field for scene representation.

Yet, traditional NeRF's representation does not reflect the fact that objects may be in different locations at different times (i.e. the scene may change along with time). Several research in NeRF tries to solve this by providing another input of time into the model [10]. However, this complicates the underlying scene representation from 3D to 4D, making the models hard to scale in big scenes. Hence will not be the first choice for dealing with moving objects in SLAM algorithms.

LiDAR-Based Semantic Segmentation methods [2] [11] [12] have shown great potential in automotive applications. Projection-based methods like SalsaNext [2] achieved real-time performances of LiDAR segmentation.

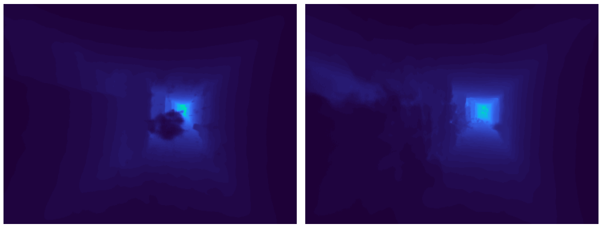

Fig . 1. Rerendered Depth of a corridor scene by LONER (left) and LONER with Semantic Segmentation (right). The left one contains artifacts due to moving people

In this project, we aim to solve the problem of moving objects in scenes by integrating LiDAR semantic segmentation models into our LONER system. Our system first processes the LiDAR scans onsite by SalsaNext [2], a real-time LiDAR point cloud semantic segmentation model, then we ignore the potentially movable objects by masking the corresponding points out when training LONER.

In the following sections, we will first introduce related works of NeRF-Based SLAM and Lidar-based segmentation in section II. Then, we will introduce our method in section III, along with some background of LONER and SalsaNext. In section IV V, we will explain our experiment design and discuss the result of the experiment. Finally, we will conclude in section VI with discuss of future works.

The upcoming sections of this report will delve into the topic at hand in the following structured manner: Section II will provide an overview of relevant research on NeRF-based SLAM methods and LiDAR-based segmentation methods. Section III will present our methodology, including an introduction to LONER and SalsaNext as background. In Section IV, we will elaborate on the design of our experiment and analyze its outcomes. Our final thoughts on the matter and future prospects will be discussed in Section V.

II. Related Works

A. NeRF-Based SLAM

iMAP [5] was the first NeRF-based SLAM method. It represents the underlying scene with a single multi-layer perceptron. The method takes RGBD images as input training data. In the tracking process, they freeze the MLP and update camera poses. Then, they jointly update the MLP and camera poses for mapping.

NICE-SLAM [6] instead represented scenes as hierarchical feature grids. This makes NICE-SLAM more scalable to large scenes.

On the other hand, NeRF-SLAM [7] uses DROID-SLAM [8] for tracking and a probabilistic NeRF [9] for scene representations.

B. LiDAR-Based Semantic Segmentation

There are many LiDAR-Based Semantic Segmentation methods [2] [11] [12] that work on the SemanticKITTI dataset [13]. LiDAR scan can be represented by a point cloud. Early methods used per-point MLP to predict the class for each point. However, their grouping and sampling algorithms are very time-consuming due to the sparse characteristic of point clouds. Other methods [2] aims to speed up the solution using a projection-based method that projects a point cloud onto a 2D plan and makes predictions based on the projected points. This achieved real-time performances although sacrificed some information during projecting. One of the state-of-the-art methods currently, 2DPASS [12], attempts to fuse information from the camera and LiDAR and make predictions based on the fused information.

C. LiDAR-Based Moving Object Segmentation

Instead of segmenting the LiDAR scan semantically (predicting movable objects), recent works start to work on the task of segmenting moving objects. LMNet [14] aims to segment moving and static objects by using the similar projection method in [2]. The model takes a range-projected point cloud and a residual image as inputs and outputs a binary mask predicting the class of each pixel in the range-projection image (moving or not).

III. Method

In this section, we first introduce two main works our project is based on. Then, we introduce the design of our new feature in LONER.

A. Background: LONER

1) Scene Representation: LONER represents the implicit scene using an MLP and a hierarchical feature grid introduced in Instant-NGP [15]. The feature grid stores trainable features in the vertices of each grid and uses linear interpolation to query the corresponding feature for each point in the space. The queried feature is then sent into the MLP as input to predict the volume density of that point. To train the network, LiDAR rays are rendered according to NeRF volume rendering equation, and backpropagation is used to update the weights of both the MLP and feature grid. For each LiDAR ray $\vec{r} = \vec{o} + t\vec{d}$, they sample N_S points along the ray, $s_i = \vec{o} + t_i\vec{d}$. Then, the weights w_i of each point are predicted by:

$$\begin{aligned} w_i &= T_i \cdot \sigma_i \\ \text{where } T_i &= \exp\Big(-\sum_{j=1}^{i-1} \sigma_j \delta_j\Big), \end{aligned} \tag{1}$$

$\delta_j = t_{j+1} - t_j$, and $\sigma_i = F(s_i; \Theta)$ is the output volume density of the nerf feature grid and MLP with parameter Θ. Instead of computing the rendered depth, they use the weights w_i directly as an input to their loss function.

2) Tracking and Mapping: Like most SLAM algorithms, LONER separates their system into tracking and mapping parts that run in parallel. In the tracking step, they estimate the relative transformation using ICP and estimate the corresponding LiDAR pose. In the mapping track, the thread receives the pose from the tracking track and determines whether to accept the scan as KeyFrame based on a temporal heuristic, then it jointly optimizes the poses of frames and a twist vector $\hat{\xi}_i \in R^6$ computed from estimated KeyFrame poses.

3) JS Divergence-based Dynamic Loss Function: In addition, LONER introduced a novel loss function based on the Jensen-Shannon divergence. For a given LiDAR ray, z^* denotes the measured depth from the LiDAR sensor. t_i denotes the training samples along the ray, and w_i represents a corresponding weight prediction from the MLP and feature grid. They define a truncated Gaussian distribution $\mathcal{K}_\epsilon$ that has a bounded domain parameterized by margin ϵ, where $\mathcal{K}_\epsilon = \mathcal{N}(0, (\epsilon/3)^2)$, as the training target distribution. And treat $w_i^* = \mathcal{K}_\epsilon(t_i - z^*)$ as the corresponding ground truth of w_i. The loss is defined as the L1 distance between the two distributions plus the opacity loss.

$$\begin{aligned} \mathcal{L}_{LOS}(\Theta) &= \|w_i^* - w_i\|_1 \\ \mathcal{L}_{opacity}(\Theta) &= \|1 - \sum_i w_i\| \\ \mathcal{L}(\Theta) &= \mathcal{L}_{LOS} + \mathcal{L}_{opacity}, \end{aligned} \tag{2}$$

In addition, they present a dynamic margin ϵ to use a larger margin for rays that have larger JS divergence with the defined Gaussian distribution. This leads to faster convergence during training.

B. Background: SalsaNext [2]

Our new feature in the system tackles the moving object problem by passing our LiDAR scans into SalsaNext to mask out movable objects. In this section, we give a brief background introduction to SalsaNext.

1) LiDAR Point Cloud Representation: To solve the sparse point problem, SalsaNext represents an unstructured 3d point cloud as a Range View image. In the 2D RV image, each LiDAR point is mapped to a 2D image coordinate using the ranged-based projection formula. The intensity value (i) and range index (r) of each LiDAR point are also stored in the RV image with separate channels, yielding a $w * h * 5$ image as input to the network.

2) Network Architecture: Following SalsaNet [16], SalsaNext takes the above RV image as input and then passes the input into an encoder-decoder architecture with several modules such as contextual module, dilated convolution, pixel-shuffle layer, dropout, and average pooling. The model predicts a class for each pixel in the image and then reprojects them onto the corresponding points.

3) Post process with kNN: To cope with the issues related to object edges when projecting predictions back to point clouds, SalsaNext employs the kNN-based post-processing process that, for each LiDAR point, they finds a subset of the close point cloud from the corresponding image pixel and pixels around it based on kNN and adjust their prediction accordingly.

C. Method Overview

To solve the problem of moving objects when running LONER, we integrate SalsaNext into the LONER codebase to mask out movable objects. As seen in Fig. 2, when the system receives a LiDAR scan, it first passes the scan through SalsaNext to get the predicted class for each point. Then, the system will delete the points that are predicted to be part of movable objects. Finally, the point cloud will be sent to the LONER model along with other information.

D. Setable Threshold

To let the user have more control over the system, we introduce a parameter threshold $gamma$ in our new feature. When a (within 0 to 1) threshold $gamma$ is provided, in addition to classifying the points with the highest-probability class, we also compute the overall probability of the point being a movable object and mask out the point if the computed probability is higher than the threshold. To elaborate, the overall probability of a point being a movable object p is defined as:

$$p = \sum_{i}^{k} p_i \tag{3}$$

Where p_i is the predicted probability of the point belonging to class i. And k is the size of our set of movable classes. We found that the threshold significantly improved the performance of our trajectory estimation by masking out any points that are possible to belong to movable objects.

IV. Experiments

A. Datasets

We evaluate our new feature on three sequences from Fusion Portable. [3] To demonstrate the effect of tackling movable objects, we selected sequences with more moving cars or moving people. The first scene we select is Corridor, a medium-scale indoor sequence of walking around a corridor with a handheld sensor. The second scene we select is Building Day, which is a medium-large-scale scene of walking from indoors to outdoors with a hand-held device. The third sequence is Campus Road, a larger outdoor scene collected by a vehicle driving on roads around campus.

B. Configurations

For different masking logic, we propose three configurations of our new feature. From masking fewer points to more points, we denote them as LONER-SAL1, LONER-SAL2, and LONER-SAL3 (LS1, LS2, LS3). Selected parameters are defined in Table I. In general, when masking fewer points, we tend to mask based on the predicted class. For instance, when the threshold is 1, we mask points only if the point is predicted to be a class that is movable. On the other hand, when masking more points, we tend to mask based on the overall probability of being a movable object instead of the predicted highest-probability class.

TABLE I
PARAMETERS FOR DIFFERENT CONFIGURATIONS.

Description	LONER	L-S1	L-S2	L-S3
Use Salsanext or not	False	True	True	True
Use kNN post-process	-	False	True	True
Masking threshold	-	1	0.05	0.01

C. Evaluation Metrics

Metrics evaluated are RMSE Absolute Trajectory Error (t_{ATE}) and relative translation error (t_{rel}) We compute our results using open-source package evo [17].

D. Performance Analysis

Our new feature offers an improvement from the original LONER system on the Campus Road scene. As shown in Table II, the new feature improved the tracking accuracy by 5.9m in the scene. In particular, by masking any points that are possible to be part of any moving object (with a probability larger than 0.01), LONER-SAL3 predicted a trajectory that is the closest (visually in Figures 2 and 3) to the ground truth on the Campus Road dataset. This shows that our settable threshold improves the trajectory predicted by our system on top of SalsaNext.

On the other hand, on the scenes Corridor and Building Day, we see similar results with or without the new feature. We believe that it is because there aren't enough moving objects in these scenes. In particular, the LONER is already doing a very good job on the scene Building Day judging by the plotted trajectory.

We conclude that our new feature can give us improved results on road scenes, which are the most important themes in the field of autonomous vehicles. In the future, we aim to try the new method in more autonomous driving scenarios.

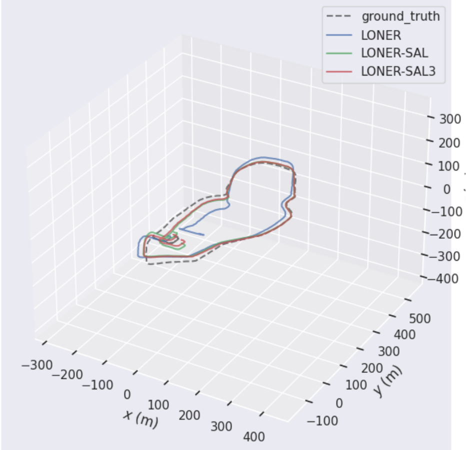

Fig . 2. Estimated trajectory of the Campus Road scene by LONER, LONER-SAL1 and LONER-SAL3

TABLE II

POSE TRACKING RESULTS ON FUSION PORTABLE SEQUENCES. t_{ATE} IS RMSE ATE (M), t_{rel} IS RMSE M RELATIVE TRANSLATION ERROR.

	Corridor		Building Day		Campus Road	
	t_{ATE}	t_{rel}	t_{ATE}	t_{rel}	t_{ATE}	t_{rel}
LONER	**13.139**	0.047	62.411	**0.039**	570.720	0.042
LONER-SAL1	13.172	0.047	62.642	**0.039**	**564.827**	**0.041**
LONER-SAL2	14.357	**0.046**	64.130	0.047	566.156	0.042
LONER-SAL3	14.218	0.047	**62.410**	0.041	567.395	0.042

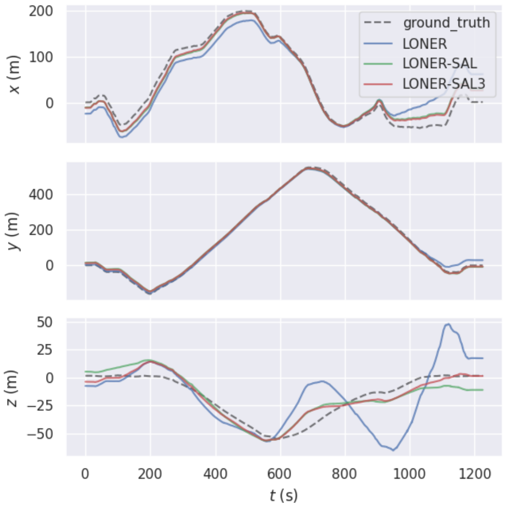

Fig . 3. Estimated trajectory of the Campus Road scene by LONER, LONER-SAL1 and LONER-SAL3 in three different dimentions

E. Runtime

On the Corridor scene, this new feature increases the overall runtime of LONER by 4.3%. Whereas in the Campus Road and Building scene, the new feature increases the overall runtime of LONER by 1.9% and 2.0%, respectively.

V. CONCLUSION

In this project, we introduced a new feature that tackles moving objects on the LONER system, the first real-time LiDAR SLAM algorithm based on NeRF representation. To tackle moving objects in the scenes during training LONER, we exploit SalsaNext, a real-time LiDAR Semantic Segmentation model based on range representation, to mask out movable objects. We designed the new feature with a tunable parameter $gamma$ that allows users to decide the amount of potential movable points to mask out during training. By testing our new feature on public datasets, we show that our new feature improves LONER in scenes with moving people and cars. However, there are still limits to our method. In particular, our method masks movable object regardless of whether it is moving or not. This may lead to loss of usable information if the objects are not actually moving.

Future works may experiment with integrating novel moving object segmentation models such as LMNet into the LONER system. However, this may involve designing a way to jointly optimize moving object detection and our LONER since moving object detection methods usually relies on having an accurate LiDAR pose. Also, we believe that nerf-based camera-lidar fusion methods may also be used on improving LiDAR semantic segmentation models by accurately fusing information from images and Lidar scans.

References

[1] "LONER: LiDAR Only NeRF for Real-Time SLAM," 2023.

[2] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving," 2020.

[3] J. Jiao, H. Wei, T. Hu, X. Hu, Y. Zhu, Z. He, J. Wu, J. Yu, X. Xie, H. Huang, R. Geng, L. Wang, and M. Liu, "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3851–3856, 2022.

[4] 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," *Commun. ACM*, vol. 65, no. 1, p. 99–106, dec 2021. [Online]. Available: https://doi-org.proxy.lib.umich.edu/10.1145/3503250

[5] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6209–6218, 2021.

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," 2022. [Online]. Available: https://arxiv.org/abs/2210.13641

[8] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," in *Advances in Neural Information Processing Systems 34 - 35th Conference on Neural Information Processing Systems, NeurIPS 2021*, ser. Advances in Neural Information Processing Systems. Neural information processing systems foundation, 2021, pp. 16 558–16 569.

[9] A. Rosinol, J. J. Leonard, and L. Carlone, "Probabilistic volumetric fusion for dense monocular slam," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, jan 2023, pp. 3096–3104. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/WACV56688.2023.00311

[10] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-NeRF: Neural Radiance Fields for Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[12] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 677–695.

[13] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[14] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data," *IEEE Robotics and Automation Letters(RA-L)*, 2021.

[15] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[16] E. E. Aksoy, S. Baci, and S. Cavdar, "Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV2020)*, 2020.

[17] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.