3D LiDAR Mapping in Dynamic Environments Using a 4D Implicit Neural Representation

Xingguang Zhong[1] Yue Pan[1] Cyrill Stachniss[1,2] Jens Behley[1]
[1]Center for Robotics, University of Bonn, [2]Lamarr Institute for Machine Learning and Artificial Intelligence
{zhong, yue.pan, cyrill.stachniss, jens.behley}@igg.uni-bonn.de

Abstract

Building accurate maps is a key building block to enable reliable localization, planning, and navigation of autonomous vehicles. We propose a novel approach for building accurate maps of dynamic environments utilizing a sequence of LiDAR scans. To this end, we propose encoding the 4D scene into a novel spatio-temporal implicit neural map representation by fitting a time-dependent truncated signed distance function to each point. Using our representation, we extract the static map by filtering the dynamic parts. Our neural representation is based on sparse feature grids, a globally shared decoder, and time-dependent basis functions, which we jointly optimize in an unsupervised fashion. To learn this representation from a sequence of LiDAR scans, we design a simple yet efficient loss function to supervise the map optimization in a piecewise way. We evaluate our approach [1] on various scenes containing moving objects in terms of the reconstruction quality of static maps and the segmentation of dynamic point clouds. The experimental results demonstrate that our method is capable of removing the dynamic part of the input point clouds while reconstructing accurate and complete 3D maps, outperforming several state-of-the-art methods.

1. Introduction

Mapping using range sensors, like LiDAR or RGB-D cameras, is a fundamental task in computer vision and robotics. Often, we want to obtain accurate maps to support downstream tasks such as localization, planning, or navigation. For achieving an accurate reconstruction of an outdoor environment, we have to account for dynamics caused by moving objects, such as vehicles or pedestrians. Furthermore, dynamic object removal plays an important role in autonomous driving and robotics applications for creating digital twins for realistic simulation and high-definition mapping, where a static map is augmented with semantic and task-relevant information.

[1]Code: https://github.com/PRBonn/4dNDF

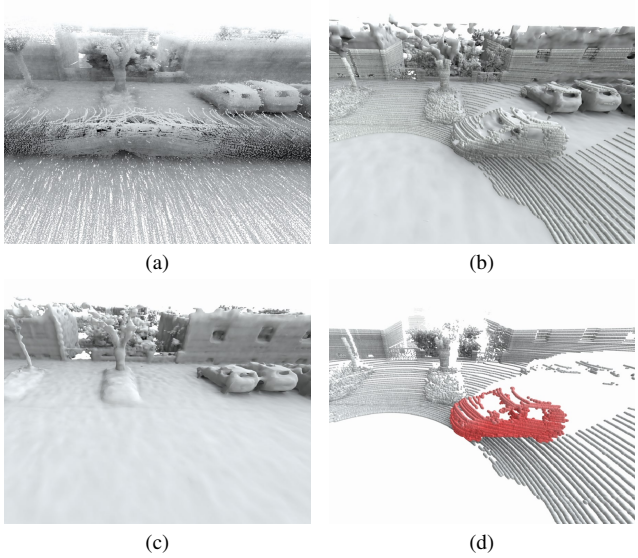


Figure 1. Given a sequence of point clouds, as shown in (a), we optimize our 4D neural representation that can be queried at arbitrary positions for a specific time. (b) Based on the estimated time-dependent TSDF values, we can extract a mesh at a specific point in time. Additionally, our 4D neural representation can be also used for static mapping (c) and dynamic object removal (d).

Mapping and state estimation in dynamic environments is a classical problem in robotics [5, 56, 57]. Approaches for simultaneous localization and mapping (SLAM) can apply different strategies to deal with dynamics. Common ways are: (1) filtering dynamics from the input [1, 30, 47, 48, 51] as a pre-processing step, which requires a semantic interpretation of the scene; (2) modeling the occupancy in the map representation [17, 34, 37, 49, 50, 64], where dynamics can be implicitly removed by retrospectively removing measurements in free space; (3) including it in the state estimation [4, 16, 55, 61, 67] to model which measurements originated from the dynamic and static parts of the environment. Our proposed method falls into the last category and allows us to model dynamics directly in the map representation leading to a spatio-temporal map representation.

Recently, implicit neural representations gained increasing interest in computer vision for novel view synthesis [35, 36] and 3D shape reconstruction [33, 40]. Due to their compactness and continuity, several approaches [65, 70, 73] investigate the use of neural representations in large-scale 3D LiDAR mapping leading to accurate maps while significantly reducing memory consumption. However, these approaches often do not address the problem of handling dynamics during mapping. The recent progress on dynamic NeRF [7, 13, 44, 52] and neural deformable object reconstruction [6, 10] indicates that neural representations can be also used to represent dynamic scenes, which inspires us to tackle the problem of mapping in dynamic environments from the perspective of 4D reconstruction.

In this paper, we propose a novel method to reconstruct large 4D dynamic scenes by encoding every point's time-dependent truncated signed distance function (TSDF) into an implicit neural scene representation. As illustrated in Fig. 1, we take sequentially recorded LiDAR point clouds collected in dynamic environments as input and generate a TSDF for each time frame, which can be used to extract a mesh using marching cubes [29]. The background TSDF, which is unchanged during the whole sequence, can be extracted from the 4D signal easily. We regard it as a static map that can be used to segment dynamic objects from the original point cloud. Compared to the traditional voxel-based mapping method, the continuous neural representation allows for the removal of dynamic objects while preserving rich map details. In summary, the main contributions of this paper are:

- We propose a novel implicit neural representation to jointly reconstruct a dynamic 3D environment and maintain a static map using sequential LiDAR scans as input.
- We employ a piecewise training data sampling strategy and design a simple, yet effective loss function that maintains the consistency of the static point supervision through gradient constraints.
- We evaluate the mapping results by the accuracy of the dynamic object segmentation as well as the quality of the reconstructed static map showing superior performance compared to several baselines. We provide our code and the data used for experiments.

2. Related Work

Mapping and SLAM in dynamic environments is a classical topic in robotics [5, 56, 57] with a large body of work, which tackles the problem by pre-processing the sensor data [1, 30, 47, 48, 51], occupancy estimation to filter dynamics by removing measurements in free space [17, 34, 37, 39, 49, 50, 64], or state estimation techniques [4, 16, 55, 61, 67]. Below, we focus on closely related approaches using neural representations but also static map building approaches for scenes containing dynamics.

Dynamic NeRF. Dynamic NeRFs aim to solve the problem of novel view synthesis in dynamic environments. Some approaches [41–43, 58, 63] address this challenge by modeling the deformation of each point with respect to a canonical frame. However, these methods cannot represent newly appearing objects. This can render them unsuited for complicated real-life scenarios. In contrast, NSFF [24] and DynIBaR [26] get rid of the canonical frame by computing the motion field of the whole scene. While these methods can deliver satisfactory results, the training time is usually in the order of hours or even days.

Another type of method leverages the compactness of the neural representation to model the 4D spatio-temporal information directly. Several works [7, 13, 52] project the 4D input into multiple voxelized lower-dimensional feature spaces to avoid large memory consumption, which improves the efficiency of the optimization. Song *et al*. [54] propose a time-dependent sliding window strategy for accumulating the voxel features. Instead of only targeting novel view synthesis, several approaches [26, 68, 71] decompose the scene into dynamic objects and static background in a self-supervised way, which inspired our work. Other approaches [22, 23, 53] accomplish neural representation-based reconstruction for larger scenes by adding additional supervision such as object masks or optical flow.

Neural representations for LiDAR scans. Recently, many approaches aim to enhance scene reconstruction using LiDAR data through neural representations. The early work URF [46] leverages LiDAR data as depth supervision to improve the optimization of a neural radiance field. With only LiDAR data as input, Huang *et al*. [20] achieve novel view synthesis for LiDAR scans with differentiable rendering. Similar to our work, Shine-mapping [73] and EIN-RUL [70] utilize sparse hierarchical feature voxel structures to achieve large-scale 3D mapping. Additionally, the data-driven approach NKSR [18] based on learned kernel regression demonstrates accurate surface reconstruction with noisy LiDAR point cloud as input. Although these approaches perform well in improving reconstruction accuracy and reducing memory consumption, none of them consider the problem of dynamic object interference in real-world environments.

Static map building and motion detection. In addition to removing moving objects from the voxel map with ray tracing, numerous works [8, 19, 31, 32] try to segment dynamic points from raw LiDAR point clouds. However, these methods require a significant amount of labeled data, which makes it challenging to generalize them to various scenarios or sensors with different scan patterns. In contrast, geometry-based, more heuristic approaches have also produced promising results. Kim *et al*. [21] solve this problem using the visibility of range images, but their results are still highly affected by the resolution. Lim *et al*. pro-

posed Erasor [27], which leverages ground fitting as prior to achieve better segmentation for dynamic points. More recent approaches [9, 28] extend it to instance level to improve results. However, these methods rely on an accurate ground fitting method, which is mainly designed for autonomous driving scenarios, which cannot be guaranteed in complex unstructured real environments.

In contrast to the approaches discussed above, we follow recent developments in neural reconstruction and propose a novel scene representation that allows us to capture the spatio-temporal progression of a scene. We represent the time-varying SDF of a scene in an unsupervised fashion, which we exploit to remove dynamic objects and reconstruct accurate meshes of the static scene.

3. Our Approach

The input of our approach is given by a sequence of point clouds, $\mathcal{S}_{1:N} = (\mathcal{S}_1, \dots, \mathcal{S}_N)$, and their corresponding global poses $\mathsf{T}_t \in \mathbb{R}^{4\times4}$, $t \in [1, N]$, estimated via scan matching, LiDAR odometry, or SLAM methods [2, 11, 12, 60]. Each scan's point cloud $\mathcal{S}_t = \{\boldsymbol{s}_t^1, \dots, \boldsymbol{s}_t^{M_t}\}$ is a set of points, $\boldsymbol{s}_t^i \in \mathbb{R}^3$, collected at time t. Given such a sequence of scans $\mathcal{S}_{1:N}$, our approach aims to reconstruct a 4D TSDF of the traversed scene and maintain a static 3D map at the same time.

In the next sections, we first introduce our spatio-temporal representation and then explain how to optimize it to represent the dynamic and static parts of a point cloud sequence $\mathcal{S}_{1:N}$.

3.1. Map Representation

The key component of our approach is an implicit neural scene representation that allows us to represent a 4D TSDF of the scene, as well as facilitates the extraction of a static map representation. Our proposed spatio-temporal scene representation is optimized for the given point cloud sequence $\mathcal{S}_{1:N}$ such that we can retrieve for an arbitrary point $\boldsymbol{p} \in \mathbb{R}^3$ and time $t \in [1, N]$ the corresponding time-varying signed distances value at that location.

Temporal representation. We utilize an TSDF to represent the scene, *i.e.*, a function that provides the signed distance to the nearest surface for any given point $\boldsymbol{p} \in \mathbb{R}^3$. The sign of the distance is positive when the point is in free space or in front of the measured surface and is negative when the point is inside the occupied space or behind the measured surface.

In a dynamic 3D scene, measuring the signed distance of any coordinate at each moment produces a time-dependent function that captures the signed distance changes over time, see Fig. 2 for an illustration. Additionally, if a coordinate is static throughout the period, the signed distance should remain constant. The key idea of our spatio-temporal scene representation is to fit the time-varying SDF

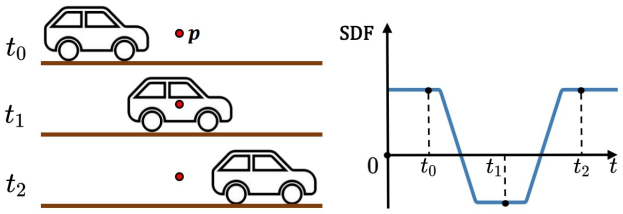


Figure 2. Principle of our 4D TSDF representation: The left figure shows a moving object and a query point $\boldsymbol{p}$. The one on the right depicts the corresponding signed distance at $\boldsymbol{p}$ over time. At t_0, $\boldsymbol{p}$'s signed distance is a positive truncated value. When the moving object reaches $\boldsymbol{p}$ at time t_1, $\boldsymbol{p}$ is inside the object and its signed distance is negative accordingly. At t_2, the moving object moved past $\boldsymbol{p}$, the signed distance of $\boldsymbol{p}$ gets positive again.

at each point with several basis functions. Inspired by Li *et al*. [26]'s representation of moving point trajectories, we exploit K globally shared basis functions $\phi_k : \mathbb{R} \mapsto \mathbb{R}$. Using these basis functions $\phi_k(t)$, we model the time-varying TSDF $F(\boldsymbol{p}, t)$ that maps a location $\boldsymbol{p} \in \mathbb{R}^3$ at time t to a signed distance as follows:

$$F(\boldsymbol{p}, t) = \sum_{k=1}^{K} w_{\boldsymbol{p}}^k \phi_k(t), \tag{1}$$

where $w_{\boldsymbol{p}}^k \in \mathbb{R}$ are optimizable location-dependent coefficients. In line with previous works [26, 62], we initialize the basis functions with discrete cosine transform (DCT) basis functions:

$$\phi_k(t) = \cos\left(\frac{\pi}{2N}(2t+1)(k-1)\right). \tag{2}$$

The first basis function for $k = 1$ is time-independent as $\phi_1(t) = 1$. During the training process, we fix $\phi_1(t)$ and determine the other basis functions by backpropagation. We consider $\phi_1(t)$'s corresponding weight $w_{\boldsymbol{p}}^1$ as the static SDF value of the point $\boldsymbol{p}$. Hence, $F(\boldsymbol{p}, t)$ consists of its static background value, *i.e.*, $w_{\boldsymbol{p}}^1 \phi_1(t) = w_{\boldsymbol{p}}^1$, and the weighted sum of dynamic basis functions $\phi_2(t), \dots, \phi_K(t)$.

As the basis functions $\phi_1(t), \dots, \phi_K(t)$ are shared between all points in the scene, we need to optimize the location-dependent weights that are implicitly represented in our spatial representation.

Spatial representation. To achieve accurate scene reconstruction while maintaining memory efficiency, we employ a multi-resolution sparse voxel grid to store spatial geometric information.

First, we accumulate the input point clouds, $\mathcal{S}_1, \dots, \mathcal{S}_N$ based on their poses $\mathsf{T}_1, \dots, \mathsf{T}_N$ computed from LiDAR odometry and generate a hierarchy of voxel grids around points to ensure complete coverage in 3D. We use a spatial hash table for fast retrieval of the resulting voxels that are only initialized if points fall into a voxel.

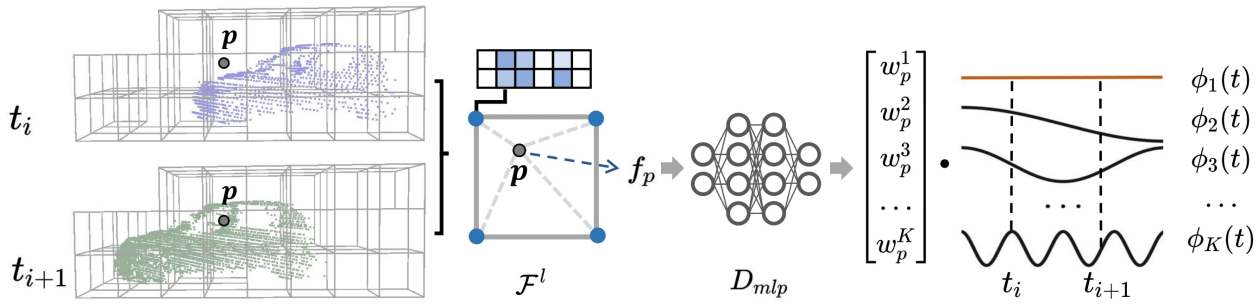


Figure 3. Overview of querying a TSDF value in our 4D map representation. For querying a point $\boldsymbol{p}$ at t_i and t_{i+1}, we first retrieve each corner's feature in $\mathcal{F}^l$ of the voxel that $\boldsymbol{p}$ is located in and obtain the fused feature $\boldsymbol{f_p}$ by trilinear interpolation. Then, we feed $\boldsymbol{f_p}$ into the decoder D_{mlp} and take the output as the weights of different basis functions $\phi_1(t), \ldots, \phi_K(t)$. Finally, we calculate the weighted sum of basis functions' values at t_i and t_{i+1} to get their respective SDF results. For simplicity, we only illustrate one level of hashed feature grids.

Similar to Instant-NGP [36], we save a feature vector $\boldsymbol{f} \in \mathbb{R}^D$ at each corner vertex of the voxel grid in each resolution level, where we denote as $\mathcal{F}^l$ the level-wise corner features. We compute the feature vector $\boldsymbol{f_p} \in \mathbb{R}^D$ for given query point $\boldsymbol{p} \in \mathbb{R}^3$ inside the hierarchical grid as follows:

$$\boldsymbol{f_p} = \sum_{l=1}^{L} \text{interpolate}(\boldsymbol{p}, \mathcal{F}^l), \tag{3}$$

where interpolate is the trilinear interpolation for a given point $\boldsymbol{p}$ using the corner features $\mathcal{F}^l$ at level l.

Then, we decode the interpolated feature vector $\boldsymbol{f_p}$ into the desired weights $\boldsymbol{w_p} = \left(w_p^1, \ldots, w_p^K\right) \in \mathbb{R}^K$ by a globally shared multi-layer perceptron (MLP) D_{mlp}:

$$\boldsymbol{w_p} = D_{\text{mlp}}(\boldsymbol{f_p}). \tag{4}$$

As every step is differentiable, we can optimize the multi-resolution feature grids $\mathcal{F}^l$, the MLP decoder D_{mlp}, and the values of the basis functions jointly by gradient descent once we have training data and corresponding target values. The SDF querying process is illustrated in Fig. 3.

3.2. Objective Function

We take samples along the rays from the input scans $\mathcal{S}_t$ to collect training data. Each scan frame $\mathcal{S}_t$ corresponds to a moment t in time, so we gather four-dimensional data points $(\boldsymbol{q}, t)$ via sampling along the ray from the scan origin $\boldsymbol{o}_t \in \mathbb{R}^3$ to a point $\boldsymbol{s}_t^i \in \mathcal{S}_t$. We can represent the sampled points $\boldsymbol{q}_s^i$ along the ray as $\boldsymbol{q}_s^i = \boldsymbol{o}_t + \lambda(\boldsymbol{s}_t^i - \boldsymbol{o}_t)$. By setting a truncation threshold τ, we split the ray into two regions, at the surface and in the free-space:

$$\mathcal{T}_{\text{surf}}^i = \left\{\boldsymbol{q}_s^i \mid \lambda \in (1-\bar{\tau}, 1+\bar{\tau})\right\} \tag{5}$$

$$\mathcal{T}_{\text{free}}^i = \left\{\boldsymbol{q}_s^i \mid \lambda \in (0, 1-\bar{\tau})\right\}, \tag{6}$$

where $\bar{\tau} = \tau \left(\|\boldsymbol{s}_t^i - \boldsymbol{o}_t\|\right)^{-1}$. Thus, $\mathcal{T}_{\text{surf}}^i$ represents the region close to the endpoint $\boldsymbol{s}_t^i \in \mathcal{S}_t$, and $\mathcal{T}_{\text{free}}^i$ is the region in the free space. We uniformly sample M_s and M_f points from $\mathcal{T}_{\text{surf}}^i$ and $\mathcal{T}_{\text{free}}^i$ separately. We obtain two sets $\mathcal{D}_{\text{surf}}$ and $\mathcal{D}_{\text{free}}$ of samples by sampling over all scans. Unlike prior work [20, 46] that use differentiable rendering to calculate the depth by integration along the ray, we design different losses for $\mathcal{D}_{\text{surf}}$ and $\mathcal{D}_{\text{free}}$ to supervise the 4D TSDF directly.

Near Surface Loss. Since the output of our 4D map is the signed distance value $\hat{d} = F(\boldsymbol{p}, t)$ at an arbitrary position $\boldsymbol{p} \in \mathbb{R}^3$ in time $t \in [1, N]$, we expect that the predicted value $\hat{d}$ does not change over time for static points. However, this cannot be guaranteed if we use the projective distance d_{surf} to the surface along the ray direction directly as the target value, since the projective distance would change over time due to the change of view direction by the moving sensor, even in a static scene. Thus, for the sampled data in $\mathcal{D}_{\text{surf}}$, *i.e.*, the sampled points near the surface, we can only obtain reliable information about the sign of the TSDF value of these points, which should be positive if the point is before the endpoint and negative if the point is behind. In addition, for a sampled point in front of the endpoint, its projective signed distance d_{surf} should be the upper bound of its actual signed distance value. And for sampled points behind the endpoint, d_{surf} should be the lower bound.

We design a piecewise loss L_{surf} to supervise the sampled points near the surface:

$$L_{\text{surf}}(\hat{d}, d_{\text{surf}}) = \begin{cases} |\,\hat{d}\,| & \text{if } \hat{d}\, d_{\text{surf}} < 0 \\ |\,\hat{d} - d_{\text{surf}}\,| & \text{if } \hat{d}\, d_{\text{surf}} > d_{\text{surf}}^2 \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $\hat{d} = F(\boldsymbol{q}, t)$ is the predicted value from our map for a sample point $\boldsymbol{q} \in \mathcal{D}_{\text{surf}}$ and d_{surf} is its corresponding projective signed distance for that sampled point in the corresponding scan $\mathcal{S}_t$. This loss punishes only a prediction when the sign is wrong or its absolute value is larger than the absolute value of d_{surf}. For a query point exactly on the surface, *i.e.*, $d_{\text{surf}} = 0$, L_{surf} is simply the L1 loss.

To calculate an accurate signed distance value and maintain the consistency of constraints for static points from different observations, we use the natural property of signed distance function to constraint the length of the gradient vector for samples inside $\mathcal{D}_{\text{surf}}$, which is called Eikonal regularization [15, 38]:

$$L_{\text{eikonal}}(\boldsymbol{p}, t) = \left(\left\| \frac{\partial F(\boldsymbol{p}, t)}{\partial \boldsymbol{p}} \right\| - 1 \right)^2, \tag{8}$$

Inspired by Neuralangelo [25], we manually add perturbations to compute more robust gradient vectors instead of using automatic differentiation, which means we compute numerical gradients:

$$\nabla_x F(\boldsymbol{p}, t) = \frac{F(\boldsymbol{p} + \boldsymbol{\epsilon}_{\boldsymbol{x}}, t) - F(\boldsymbol{p} - \boldsymbol{\epsilon}_{\boldsymbol{x}}, t)}{2\epsilon}, \tag{9}$$

where $\nabla_x F(\boldsymbol{p}, t)$ is the component of the gradient $\frac{\partial F(\boldsymbol{p}, t)}{\partial \boldsymbol{p}}$ on the x axis, and $\boldsymbol{\epsilon}_{\boldsymbol{x}} = (\epsilon, 0, 0)^\top$ is the added perturbation. We apply the same operation on y and z axes to calculate the numerical gradient. Furthermore, in order to get faster convergence at the beginning and ultimately recover the rich geometric details, we first set a large ϵ and gradually reduce it during the training process.

Free Space Loss. As we tackle the problem of mapping in dynamic environments, we cannot simply accumulate point clouds and then calculate accurate supervision of signed distance value via nearest neighbor search. Therefore, we use a L1 loss L_{free} to constrain the signed distance prediction $\hat{d}$ of the free space points, *i.e.*, $\boldsymbol{p} \in \mathcal{D}_{\text{free}}$:

$$L_{\text{free}}(\hat{d}) = |\hat{d} - \tau|, \tag{10}$$

where τ is the truncation threshold we used in Sec. 3.2.

Thanks to our spatio-temporal representation, a single query point can get both, static and dynamic TSDF values. Thus, for some regions that are determined to be free space, we can directly add constraints to their static TSDF values.

We divide the free space points $\mathcal{D}_{\text{free}}$ into dense and sparse subset $\mathcal{D}_{\text{dense}}$ and $\mathcal{D}_{\text{sparse}}$ based on a threshold r_{dense} for the distance from the free space point sampled at time t to the scan origin $\boldsymbol{o}_t$. For each point $\boldsymbol{p} \in \mathcal{D}_{\text{dense}}$, we find the nearest neighbor $\boldsymbol{n_p}$ in the corresponding scan $\mathcal{S}_t$, *i.e.*, $\boldsymbol{n_p} = \arg\min_{\boldsymbol{q} \in \mathcal{S}_t} \|\boldsymbol{p} - \boldsymbol{q}\|_2$. Let $\mathcal{D}_{\text{certain}} = \{\boldsymbol{p} \in \mathcal{D}_{\text{dense}} \mid ||\boldsymbol{p} - \boldsymbol{n_p}|| > \tau\}$ be the points that we consider in the certain free space. Then, we supervise $\boldsymbol{p} \in \mathcal{D}_{\text{certain}}$ by its static signed distance value directly:

$$L_{\text{certain}}(\boldsymbol{p}) = |w^1_{\boldsymbol{p}} - \tau|, \tag{11}$$

where $w^1_{\boldsymbol{p}}$ is the first weight of the decoder's output.

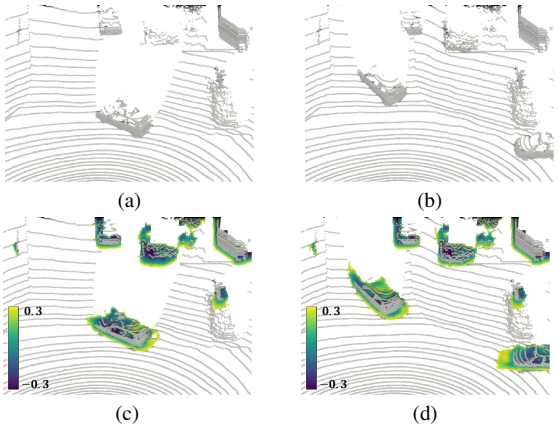


Figure 4. Reconstructed TSDF for KITTI dataset [14]: Subfigures (a) and (b) are the input neighboring frames. Correspondingly, (c) and (d) are horizontal TSDF slices queried from our 4D map. Note that we only display the TSDF values that are less than 0.3m.

In summary, the final loss L_{total} is given by:

$$\begin{aligned} L_{\text{total}} = & \frac{1}{|\mathcal{D}_{\text{surf}}|} \sum_{(\boldsymbol{p},t) \in \mathcal{D}_{\text{surf}}} L_{\text{surf}}(\hat{d}, d_{\text{surf}}) + \lambda_e L_{\text{eikonal}}(\boldsymbol{p}, t) \\ & + \frac{\lambda_f}{|\mathcal{D}_{\text{free}}|} \sum_{(\boldsymbol{p},t) \in \mathcal{D}_{\text{free}}} L_{\text{free}}(\hat{d}) \\ & + \frac{\lambda_c}{|\mathcal{D}_{\text{certain}}|} \sum_{(\boldsymbol{p},t) \in \mathcal{D}_{\text{certain}}} L_{\text{certain}}(\boldsymbol{p}), \end{aligned} \tag{12}$$

where $\hat{d} = F(\boldsymbol{p}, t)$ is the predicted signed distance at the sample position $\boldsymbol{p}$ at time t and d_{surf} is the projective signed distance of sample $\boldsymbol{p}$. With the above loss function and data sampling strategy, we train our map offline until convergence. In Fig. 4, we show TSDF slices obtained using our optimized 4D map at different times.

One application of our 4D map representation is dynamic object segmentation. For a point $\boldsymbol{p}$ in the input scans $\mathcal{S}_{1:N}$, its static signed distance value $w^1_{\boldsymbol{p}}$ can be obtained by a simple query. If $\boldsymbol{p}$ belongs to the static background, it should have $w^1_{\boldsymbol{p}} = 0$. Therefore, we simply set a threshold d_{static} and regard a point as dynamic if $w^1_{\boldsymbol{p}} > d_{\text{static}}$.

3.3. Implementation Details

As hyperparameters of our approach, we use the values listed in Tab. 1 in all LiDAR experiments. Additional parameters are determined by the characteristics of the sensor and the dimensions of the scene. For instance, in the reconstruction of autonomous driving scenes, like KITTI, we set the highest resolution for the feature voxels to $0.3\,\text{m}$. The truncation distance is set to $\tau = 0.5\,\text{m}$, and the dense area split threshold $r_{\text{dense}} = 15\,\text{m}$. Regarding training time, it takes 12 minutes to train 140 frames from the KITTI dataset using a single Nvidia Quadro RTX 5000.

4. Experiments

In this section, we show the effectiveness of our proposed approach with respect to two aspects: (1) Static mapping quality: The static TSDF built by our method allows us to extract a surface mesh using marching cubes [29]. We compare this extracted mesh with the ground truth mesh to evaluate the reconstruction. (2) Dynamic object segmentation: As mentioned above, our method can segment out the dynamic objects in the input scans. We use point-wise dynamic object segmentation accuracy to evaluate the results.

4.1. Static Mapping Quality

Datasets. We select two datasets collected in dynamic environments for quantitative evaluation. One is the synthetic dataset *ToyCar3* from Co-Fusion [47], which provides accurate depth images and accurate masks of dynamic objects rendered using Blender, but also depth images with added noise. For this experiment, we select 150 frames from the whole sequence, mask out all dynamic objects in the accurate depth images, and accumulate background static points as the ground-truth static map. The original noisy depth images are used as the input for all methods.

Furthermore, we use the *Newer College* [45] dataset as the real-world dataset, which is collected using a 64-beam LiDAR. Compared with synthetic datasets, it contains more uncertainty from measurements and pose estimates. We select 1,300 frames from the courtyard part for testing and this data includes a few pedestrians as dynamic objects. This dataset offers point clouds obtained by a high-precision terrestrial laser scanner that can be directly utilized as ground truth to evaluate the mapping quality.

Metric and Baselines. We report the reconstruction accuracy, completeness, the Chamfer distance, and the F1-score. Further details on the computation of the metrics can be found in the supplement.

We compare our method with several different types of state-of-the-art methods: (i) the traditional TSDF-fusion method, VDBfusion [59], which uses space carving to eliminate the effects of dynamic objects, (ii) the data-driven-based method, neural kernel surface reconstruction (NKSR) [18], and (iii) the neural representation based 3D mapping approach, SHINE-mapping [73].

For NKSR [18], we use the default parameters provided by Huang *et al*. with their official implementation. To ensure a fair comparison with SHINE-mapping, we adopt an equal number of free space samples (15 samples), aligning with our method for consistency.

For the *ToyCar3* dataset, we set VDB-Fusion's resolution to 1 cm. To have all methods with a similar memory consumption, we set the resolution of SHINE-mapping's leaf feature voxel to 2 cm, and our method's highest resolution accordingly to 2 cm. For the *Newer College* dataset, we set the resolution to 10 cm, 30 cm, and 30 cm respectively.

Table 1. Hyperparameters of our approach.

Parameter	Value	Description
L	2	number of feature voxels level
D	8	The length of feature vectors
K	32	The number of basis functions
D_{mlp}	2×64	layer and size of the MLP decoder
M_s	5	The number of surface area samples
M_f	15	The number of free space samples
λ_e	0.02	weight for Eikonal loss
λ_f	0.25	weight for free space loss
λ_c	0.2	weight for certain free loss

Table 2. Quantitative results of the reconstruction quality on *ToyCar3*. We report the distance error metrics, namely completion, accuracy and Chamfer-L1 in cm. Additionally, we show the F-score in % with a 1 cm error threshold.

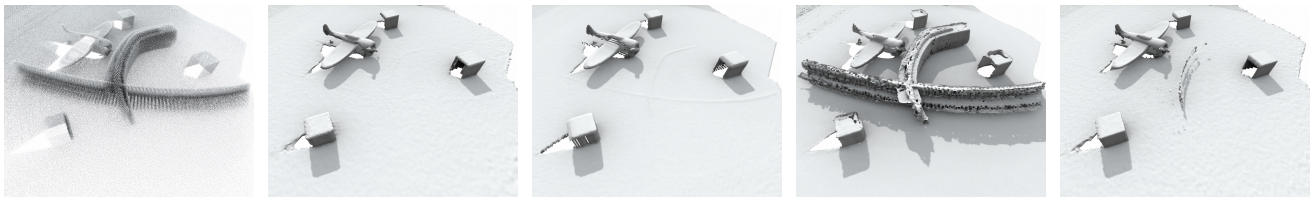
Method	Comp. ↓	Acc. ↓	C-L1 ↓	F-score ↑
VDB-fusion [59]	0.574	0.481	0.528	97.95
NKSR [18]	0.526	2.809	1.667	89.54
SHINE-mapping [73]	0.583	0.626	0.605	98.01
Ours	**0.438**	**0.468**	**0.452**	**98.35**

Results. The quantitative results for synthetic dataset *ToyCar3* and real-world dataset *Newer College* are presented in Tab. 2 and Tab. 3, respectively. We also show the extracted meshes from all methods in Fig. 5 and Fig. 6.

Our method outperforms the baselines in terms of Completeness and Chamfer distance for both datasets (*cf*. Fig. 5 and Fig. 6). Regarding the accuracy, SHINE-mapping and VDB-Fusion can filter part of high-frequency noise by fusion of multiple frames, resulting in better performance on noisy *Newer College* dataset. In comparison, our method considers every scan as accurate to store 4D information, which makes it more sensitive to measurement noise. On the *ToyCar3* dataset, both our method and VDB-Fusion successfully eliminate all moving objects. However, on the *Newer College* dataset, VDB-Fusion incorrectly eliminates the static tree and parts of the ground, resulting in poor completeness shown in Tab. 3. SHINE-mapping eliminates dynamic pedestrians on the *Newer College* dataset but retains a portion of the dynamic point cloud on the *ToyCar3* dataset, which has a larger proportion of dynamic objects, leading to poorer accuracy in Tab. 2. NKSR performs the worst accuracy because it is unable to eliminate dynamic objects, which means it's not suitable to apply NKSR in dynamic real-world scenes directly.

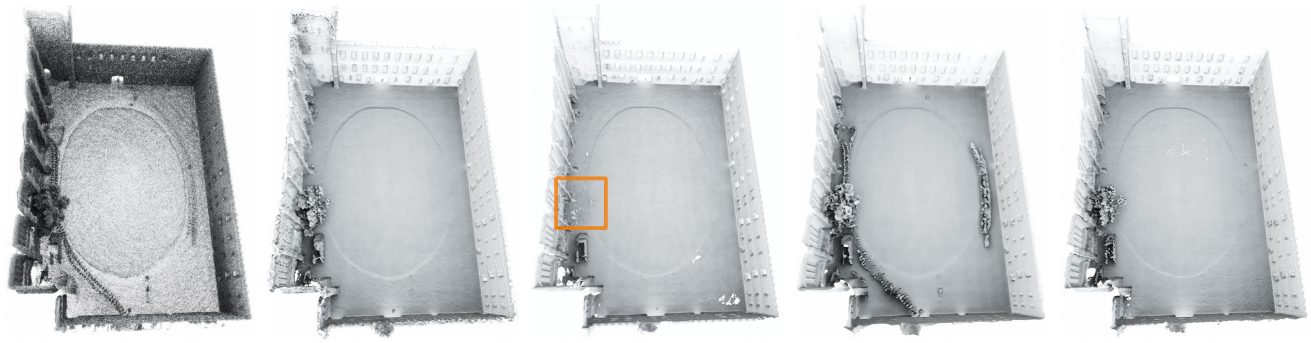
4.2. Dynamic Object Segmentation

Datasets. For dynamic object segmentation, we use the KTH-Dynamic-Benchmark [72] for evaluation, which in-



(a) Merged input scans (b) Ours (c) VDB-Fusion [59] (d) NKSR [18] (e) SHINE-mapping [73]

Figure 5. A comparison of the static mapping results of different methods on the *ToyCar3* dataset. There are two dynamic toy cars moving through the scene. Our method can reconstruct the static scene with fine details and eliminate the dynamic car.



(a) Merged input scans (b) Ours (c) VDB-Fusion [59] (d) NKSR [18] (e) SHINE-mapping [73]

Figure 6. A comparison of the static mapping results of different methods on the *Newer College* dataset. Several pedestrians are moving through the scene during the data collection. Our method can reconstruct the static scene completely and eliminate the moving pedestrians. Although VDB-Fusion manages to eliminate the pedestrians, it incorrectly removes the tree highlighted in the orange box.

Table 3. Quantitative results of the reconstruction quality on *Newer College*. We report the distance error metrics, namely completion, accuracy and Chamfer-L1 in cm. Additionally, we show the F-score in % with a 20 cm error threshold.

Method	**Comp.** ↓	**Acc.** ↓	**C-L1** ↓	**F-score** ↑
VDB-fusion [59]	7.32	5.99	6.65	96.68
NKSR [18]	6.87	9.28	8.08	95.65
SHINE-mapping [73]	6.80	**5.86**	6.33	**97.67**
Ours	**5.85**	6.49	**6.17**	97.50

cludes four sequences in total: sequence 00 (frame 4,390 – 4,530) and sequence 05 (frame 2,350 – 2,670) from the KITTI dataset [3, 14], which are captured by a 64-beam LiDAR, one sequence from the Argoverse2 dataset [66] consisting of 575 frames captured by two 32-beam LiDARs, and a semi-indoor sequence captured by a sparser 16-beam LiDAR. All sequences come with corresponding pose files and point-wise dynamic or static labels as the ground truth. It is worth noting that the poses for KITTI 00 and 05 were obtained from SuMa [2] and the pose files for the Semi-indoor sequence come from NDT-SLAM [50].

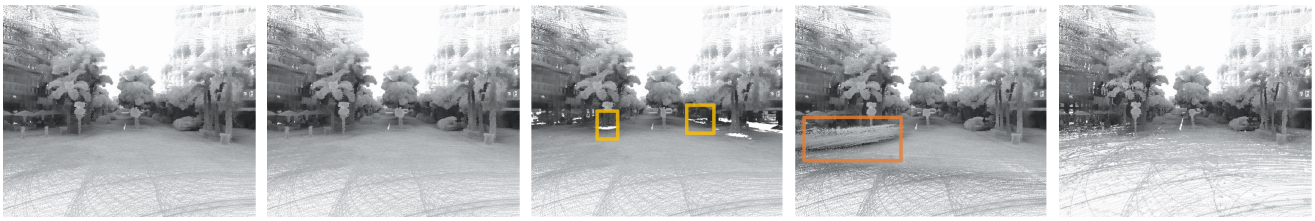
Metric and Baselines. The KTH-Dynamic-Benchmark evaluates the performance of the method by measuring the classification accuracy of dynamic points (DA%), static points (SA%) and also their associated accuracy (AA%) where $AA = \sqrt{DA \cdot SA}$. The benchmark provides various baselines such as the state-of-the-art LiDAR dynamic object removal methods – Erasor [27] and Removert [21], as well as the traditional 3D mapping method, Octomap [17, 69], and its modified versions, Octomap with ground fitting and outlier filtering. As SHINE-mapping demonstrates the ability to remove dynamic objects in our static mapping experiments, we also report its result in this benchmark. Additionally, we report the performance of the state-of-the-art online moving object segmentation methods, 4DMOS [31] and its extension MapMOS [32]. As these two methods utilize KITTI sequences 00 and 05 for training, we only show the results of the remaining two sequences. For the parameter setting, we set our method's leaf resolution to 0.3 m, and the threshold for segmentation as $d_{\text{static}} = 0.16$ m. We set the leaf resolution for Octomap to 0.1 m.

Results. The quantitative results of the dynamic object segmentation are shown in Tab. 4. And we depict the accumulated static points generated by different methods in Fig. 7. We can see that our method achieves the best associated accuracy (AA) in three autonomous driving sequences (KITTI 00, KITTI 05, Argoverse2) and vastly outperforms baselines. The supervised learning-based methods 4DMOS and MapMOS do not obtain good dynamic accuracy (DA) due to limited generalizability. Erasor and Octomap tend to over-segment dynamic objects, resulting in poor static accuracy (SA). Removert and SHINE-mapping are too conservative and cannot detect all dynamic objects. Benefiting

Table 4. Quantitative results of the dynamic object removal quality on the KTH-Dynamic-Benchmark. We report the static accuracy SA, dynamic static DA and the associated accuracy AA. Octomap* refers to the modified Octomap implementation by Zhang *et al.* [72].

	KITTI Seq. 00			KITTI Seq. 05			Argoverse2			Semi-Indoor		
Method	**SA**	**DA**	**AA**	**SA**	**DA**	**AA**	**SA**	**DA**	**AA**	**SA**	**DA**	**AA**
Octomap [17]	68.05	**99.69**	82.37	66.28	**99.24**	81.10	65.91	96.70	79.84	88.97	**82.18**	**85.51**
Octomap* [72]	93.06	98.67	95.83	93.54	92.48	93.01	82.66	82.44	82.55	96.79	73.50	84.34
Removert [21]	99.44	41.53	64.26	99.42	22.28	47.06	98.97	31.16	55.53	99.96	12.15	34.85
Erasor [27]	66.70	98.54	81.07	69.40	99.06	82.92	77.51	**99.18**	87.68	94.90	66.26	79.30
SHINE [73]	98.99	92.37	95.63	98.91	53.27	72.58	97.66	72.62	84.21	98.88	59.19	76.51
4DMOS [31]	-	-	-	-	-	-	99.94	69.33	83.24	**99.99**	10.60	32.55
MapMOS [32]	-	-	-	-	-	-	**99.96**	85.88	92.65	**99.99**	4.75	21.80
Ours	**99.46**	98.47	**98.97**	**99.54**	98.36	**98.95**	99.17	95.91	**97.53**	94.17	72.79	82.79

(a) Ground truth (b) Ours (c) Erasor [27] (d) Removert [21] (e) Octomap* [72]



(f) Ground truth (g) Ours (h) Erasor [27] (i) Removert [21] (j) Octomap* [72]

Figure 7. Comparison of dynamic object removal results produced by our proposed method and three baseline methods on the Argoverse2 data sequence of the KTH-benchmark. We show the bird's eye view on the first row and the zoomed view from the blue frustum shown in (a) on the second row. For the ground truth results in (a), the dynamic objects are shown in red. We only show the static points of ground truth for clearer comparison in zoomed view (f). We highlight the over-segmented parking car and sign by Erasor and the undetected moving vehicle by Removert.

from the continuity and large capacity of the 4D neural representation, we strike a better balance between preserving static background points and removing dynamic objects.

It is worth mentioning again that our method does not rely on any pre-processing or post-processing algorithm such as ground fitting, outlier filtering, and clustering, but also does not require labels for training.

5. Conclusion

In this paper, we propose a 4D implicit neural map representation for dynamic scenes that allows us to represent the TSDF of static and dynamic parts of a scene. For this purpose, we use a hierarchical voxel-based feature representation that is then decoded into weights for basis functions to represent a time-varying TSDF that can be queried at arbitrary locations. For learning the representation from a sequence of LiDAR scans, we design an effective data sampling strategy and loss functions. Equipped with our proposed representation, we experimentally show that we are able to tackle the challenging problems of static mapping and dynamic object segmentation. More specifically, our experiments show that our method has the ability to accurately reconstruct 3D maps of the static parts of a scene and can completely remove moving objects at the same time.

Limitations. While our method achieves compelling results, we have to acknowledge that we currently rely on estimated poses by a separate SLAM approach, but also cannot apply our approach in an online fashion. However, we see this as an avenue for future research into joint incremental mapping and pose estimation.

Acknowledgements. We thank Benedikt Mersch for the fruitful discussion and for providing experiment baselines.

References

[1] Ioan A. Barsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust Dense Mapping for Large-Scale Dynamic Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 1, 2

[2] Jens Behley and Cyrill Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018. 3, 7

[3] Jens Behley, Martin Garbade, Aandres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019. 7

[4] Peter Biber and Tom Duckett. Dynamic Maps for Long-Term Operation of Mobile Service Robots. In *Proc. of Robotics: Science and Systems (RSS)*, 2005. 1, 2

[5] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Trans. on Robotics (TRO)*, 32(6):1309–1332, 2016. 1, 2

[6] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[7] Ang Cao and Justin Johnson. HexPlane: A Fast Representation for Dynamic Scenes. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[8] Xieyuanli Chen, Shijie Li, Benedikt Mersch, Louis Wiesmann, Juergen Gall, Jens Behley, and Cyrill Stachniss. Moving Object Segmentation in 3D LiDAR Data: A Learning-based Approach Exploiting Sequential Data. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):6529–6536, 2021. 2

[9] Xieyuanli Chen, Benedikt Mersch, Lucas Nunes, Rodrigo Marcuzzi, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Automatic Labeling to Generate Training Data for Online LiDAR-Based Moving Object Segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):6107–6114, 2022. 3

[10] Xu Chen, Tianjian Jiang, Jie Song, Max Rietmann, Andreas Geiger, Michael J. Black, and Otmar Hilliges. Fast-snarf: A fast deformer for articulated neural fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(10): 11796–11809, 2023. 2

[11] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and Francois Goulette. CT-ICP Real-Time Elastic LiDAR Odometry with Loop Closure. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022. 3

[12] Jean-Emmanuel Deschaud. IMLS-SLAM: scan-to-model matching based on 3D data. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018. 3

[13] Sara Fridovich-Keil, Giacomo Meanti, Frederik R. Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[14] Andreas Geiger, Peter Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5, 7

[15] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit Geometric Regularization for Learning Shapes. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, 2020. 5

[16] Dirk Hähnel, Dirk Schulz, and Wolfram Burgard. Mobile robot mapping in populated environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2002. 1, 2

[17] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34(3):189–206, 2013. 1, 2, 7, 8

[18] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural Kernel Surface Reconstruction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6, 7

[19] Shengyu Huang, Zan Gojcic, Jiahui Huang, Andreas Wieser, and Konrad Schindler. Dynamic 3D Scene Analysis by Point Cloud Accumulation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2022. 2

[20] Shengyu Huang, Zan Gojcic, Zian Wang, Francis Williams, Yoni Kasten, Sanja Fidler, Konrad Schindler, and Or Litany. Neural LiDAR Fields for Novel View Synthesis. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023. 2, 4

[21] Giseop Kim and Ayoung Kim. Remove, Then Revert: Static Point Cloud Map Construction Using Multiresolution Range Images. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020. 2, 7, 8

[22] Xin Kong, Shikun Liu, Marwan Taher, and Andrew J. Davison. vMAP: Vectorised Object Mapping for Neural Field SLAM. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[23] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[24] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[25] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5

[26] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DynIBaR: Neural Dynamic Image-Based Rendering. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[27] Hyungtae Lim, Sungwon Hwang, and Hyun Myung. ERASOR: Egocentric Ratio of Pseudo Occupancy-Based Dynamic Object Removal for Static 3D Point Cloud Map Building. *IEEE Robotics and Automation Letters (RA-L)*, 6(2): 2272–2279, 2021. 3, 7, 8

[28] Hyungtae Lim, Lucas Nunes, Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. ERASOR2: Instance-Aware Robust 3D Mapping of the Static World in Dynamic Scenes. In *Proc. of Robotics: Science and Systems (RSS)*, 2023. 3

[29] William E. Lorensen and Harvey E. Cline. Marching Cubes: a High Resolution 3D Surface Construction Algorithm. In *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1987. 2, 6

[30] John McCormac, Ankur Handa, Aandrew J. Davison, and Stefan Leutenegger. SemanticFusion: Dense 3D Semantic Mapping with Convolutional Neural Networks. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017. 1, 2

[31] Benedikt Mersch, Xieyuanli Chen, Ignacio Vizzo, Lucas Nunes, Jens Behley, and Cyrill Stachniss. Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7503–7510, 2022. 2, 7, 8

[32] Benedikt Mersch, Tiziano Guadagnino, Xieyuanli Chen, Tiziano, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Building Volumetric Beliefs for Dynamic Environments Exploiting Map-Based Moving Object Segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):5180–5187, 2023. 2, 7, 8

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[34] Daniel Meyer-Delius, Maximilitan Beinhofer, and Wolfram Burgard. Occupancy Grid Models for Robot Mapping in Changing Environments. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2012. 1, 2

[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2020. 2

[36] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 41(4): 102:1–102:15, 2022. 2, 4

[37] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1, 2

[38] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. isdf: Real-time neural signed distance fields for robot perception. In *Proc. of Robotics: Science and Systems (RSS)*, 2022. 5

[39] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019. 2

[40] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[41] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable Neural Radiance Fields. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021. 2

[42] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. on Graphics (TOG)*, 40(6), 2021.

[43] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[44] Sameera Ramasinghe, Violetta Shevchenko, Gil Avraham, and Anton Van Den Hengel. Blirf: Band limited radiance fields for dynamic scene modeling. *arXiv preprint arXiv:2302.13543*, 2023. 2

[45] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020. 6

[46] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 4

[47] Martin Rünz and Lourdes Agapito. Co-Fusion: Real-Time Segmentation, Tracking and Fusion of Multiple Objects. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017. 1, 2, 6

[48] Martin Rünz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, 2018. 1, 2

[49] Jari Saarinen, Henrik Andreasson, and Achim Lilienthal. Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012. 1, 2

[50] Jari P. Saarinen, Todor Stoyanov, Henrik Andreasson, and Achim J. Lilienthal. Fast 3D Mapping in Highly Dynamic

Environments Using Normal Distributions Transform Occupancy Maps. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013. 1, 2, 7

[51] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H. Kelly, and Andrew J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2

[52] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d : Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[53] Chonghyuk Song, Gengshan Yang, Kangle Deng, Jun-Yan Zhu, and Deva Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2023. 2

[54] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. NeRFPlayer: A Streamable Dynamic Scene Representation with Decomposed Neural Radiance Fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 2

[55] Cyrill Stachniss and Wolfram Burgard. Mobile Robot Mapping and Localization in Non-Static Environments. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2005. 1, 2

[56] Cyrill Stachniss, John J. Leonard, and Sebastian Thrun. *Springer Handbook of Robotics, 2nd edition*, chapter Chapt. 46: Simultaneous Localization and Mapping. Springer Verlag, 2016. 1, 2

[57] Sebstian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005. 1, 2

[58] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021. 2

[59] Ignacio Vizzo, Tiziano Guadagnino, Jens Behley, and Cyrill Stachniss. VDBFusion: Flexible and Efficient TSDF Integration of Range Sensor Data. *Sensors*, 22(3):1296, 2022. 6, 7

[60] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill 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 (RA-L)*, 8(2):1029–1036, 2023. 3

[61] Aishan Walcott-Bryant, Michael Kaess, Hordur Johannsson, and John J. Leonard. Dynamic Pose Graph SLAM: Long-Term Mapping in Low Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012. 1, 2

[62] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021. 3

[63] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-Viewpoint Rendering of Moving People From Monocular Video. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[64] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, Ben Glocker, and Andrew J. Davison. ElasticFusion: Dense SLAM Without A Pose Graph. In *Proc. of Robotics: Science and Systems (RSS)*, 2015. 1, 2

[65] Louis Wiesmann, Tiziano Guadagnino, Ignacio Vizzo, Nicky Zimmerman, Yue Pan, Haofei Kuang, Jens Behley, and Cyrill Stachniss. LocNDF: Neural Distance Field Mapping for Robot Localization. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):4999–5006, 2023. 2

[66] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2021. 7

[67] Denis F. Wolf and Guarav S. Sukhatme. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. *Autonomous Robots*, 19, 2005. 1, 2

[68] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. Dˆ2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[69] Kai M. Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems. In *Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation, IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010. 7

[70] Dongyu Yan, Xiaoyang Lyu, Jieqi Shi, and Yi Lin. Efficient Implicit Neural Reconstruction Using LiDAR. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. 2

[71] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[72] Qingwen Zhang, Daniel Duberg, Ruoyu Geng, Mingkai Jia, Lujia Wang, and Patric Jensfelt. A dynamic points removal benchmark in point cloud maps. In *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 608–614, 2023. 6, 8

[73] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. 2, 6, 7, 8

Certificate of Reproducibility

The authors of this publication declare that:

1. The software related to this publication is distributed in the hope that it will be useful, support open research, and simplify the reproducability of the results but it comes without any warrenty and without even the implied warranty of merchantability or fitness for a particular purpose.

2. *Xingguang Zhong* primarily developed the implementation related to this paper. This was done on Ubuntu 22.04.

3. *Jens Behley* verified that the code can be executed on a machine that follows the software specification given in the Git repository available at:

   `https://github.com/PRBonn/4dNDF`

4. *Jens Behley* verified that the experimental results presented in this publication can be reproduced using the implementation used at submission, which is labeled with a tag in the Git repository and can be retrieved using the command:

   ```
   git checkout zhong2024cvpr
   ```

3D LiDAR Mapping in Dynamic Environments Using a 4D Implicit Neural Representation

Supplementary Material

In this supplementary material, we present ablation studies that explore the impact of combining different types of losses and changing the number of basis functions. We also provide additional details on the implementation and experiments we conducted. Moreover, we showcase more qualitative results for dynamic object segmentation. We also open-source the scripts and code needed to reproduce all the experiments of our paper, which can be found at https://github.com/PRBonn/4dNDF

A. Ablation Study

The experiments on static mapping quality focus more on the result of the surface reconstruction. In order to study the reconstruction performance in the free space, we chose the KITTI seq. 05 from KTH-Dynamic-benchmark for the ablation study.

Influence of loss terms. Tab. 1 presents the impact of different combinations of losses influences the segmentation result. Note that we refer to the settings by the letters (A)–(F) in the first column of Tab. 1 in the following discussion. From the table, we can observe that only enabling L_{free}, *i.e.* case (A), alone results in a low score for DA, indicating that a large portion of dynamic objects remain unsegmented. However, adding of L_{eikonal} simultaneously, *i.e.*, case (B), can enhance the result. In case (C), the majority of dynamic objects can be eliminated by applying L_{certain}. However, since L_{certain} is only applied in the densely observed area, split by a hyperparameter r_{dense}, some distant dynamic points are preserved. By adding L_{free}, this problem can be solved, leading to further improvement in the result. In Fig. 1, we show qualitatively the influence of the different parts of the loss.

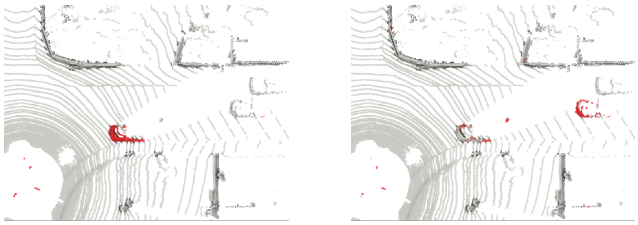
Influence of number of basis functions. Tab. 2 shows the impact of using different numbers of basis functions (K) on dynamic object segmentation result. $K = 1$ implies the map degenerates to 3D, which means the output of D_{mlp} is a single value representing time-independent signed distance. This leads to poor performance of both AA and DA. Increasing the value of K has a direct impact on the map's capacity, resulting in more accurate dynamic point segmentation. The optimal performance is achieved when K is set to 32. Further increasing the value of K does not lead to diminishing returns in terms of performance or even a degradation of the results. Therefore, we select $K = 32$ as the number of basis functions for all of our experiments in the main paper.

Table 1. Ablation study of losses combination on KITTI seq. 05 sequence from KTH-Dynamic-benchmark

	L_{free}	L_{certain}	L_{eikonal}	SA	DA	AA
A	✓			99.71	45.05	67.02
B	✓		✓	99.58	57.68	75.79
C		✓		99.96	89.11	95.38
D		✓	✓	99.11	92.42	95.71
E	✓	✓		99.44	99.50	97.45
F	✓	✓	✓	99.54	98.36	**98.95**

Table 2. Ablation study of the number of basis functions (K) on KITTI seq. 05 sequence from KTH-Dynamic-benchmark

K	SA	DA	AA
1	91.82	52.59	69.49
4	97.73	95.30	96.51
8	99.55	95.71	97.61
16	97.60	98.06	97.83
24	99.14	97.68	98.41
32	99.54	98.36	**98.95**
40	96.78	98.57	97.67
48	99.67	98.02	98.84



(a) wo L_{free} (b) wo L_{certain}

(c) full losses (d) ground truth

Figure 1. Segmentation results on one frame in the ablation study for losses. (a) shows the result when we turn off L_{free}, corresponding to case (D) in Tab. 1, we can see that the moving car far away from the sensor is not segmented. And turning off L_{certain} (case (B)) leads to poor dynamics segmentation in the dense observed area, which can be seen in (b).

B. Further Details on Experiments

B.1. Static Mapping Quality

Metrics. We represent the points sampled from ground-truth mesh or point clouds as P_{gt}, and represent the points sampled from estimated mesh as P_{es}, as detailed below for the different datasets. Then, we calculate the metrics as below:

$$\text{Comp} = \frac{1}{|P_{\text{gt}}|} \sum_{\mathbf{p}_{\text{gt}} \in P_{\text{gt}}} \min_{\mathbf{p}_{\text{es}} \in P_{\text{es}}} (\|\mathbf{p}_{\text{gt}} - \mathbf{p}_{\text{es}}\|), \tag{1}$$

$$\text{Acc.} = \frac{1}{|P_{\text{es}}|} \sum_{\mathbf{p}_{\text{es}} \in P_{\text{es}}} \min_{\mathbf{p}_{\text{gt}} \in P_{\text{gt}}} (\|\mathbf{p}_{\text{es}} - \mathbf{p}_{\text{gt}}\|), \tag{2}$$

$$\text{C-L1} = \frac{1}{2} (\text{Comp.} + \text{Acc.}), \tag{3}$$

$$\text{Precision} = \frac{\left|\left\{\mathbf{p}_{\text{es}} \in P_{\text{es}} \mid \min_{\mathbf{p}_{\text{gt}} \in P_{\text{gt}}} \|\mathbf{p}_{\text{es}} - \mathbf{p}_{\text{gt}}\| < \xi\right\}\right|}{|P_{\text{es}}|}, \tag{4}$$

$$\text{Recall} = \frac{\left|\left\{\mathbf{p}_{\text{gt}} \in P_{\text{gt}} \mid \min_{\mathbf{p}_{\text{es}} \in P_{\text{es}}} \|\mathbf{p}_{\text{gt}} - \mathbf{p}_{\text{es}}\| < \xi\right\}\right|}{|P_{\text{gt}}|}, \tag{5}$$

We report completeness (Comp.), accuracy (Acc.), Chamfer-Distance (C-L1) and F-score in the main text. For F-score, we use $\xi = 0.1\,\text{cm}$ in *ToyCar3* dataset and $\xi = 20\,\text{cm}$ in the real-world *Newer College* dataset.

Experiment settings. For *ToyCar3* dataset, we downsample the accumulated background point cloud with a resolution of $0.5\,\text{cm}$ and use the resulting point cloud as P_{gt}. We then uniformly sample the same number of points as P_{gt} on the mesh obtained by the methods and consider it as P_{es}. We used a resolution of $0.5\,\text{cm}$ for marching cubes to extract meshes in both our method and SHINE-mapping [4].

For *Newer College* dataset, we directly use the ground-truth point cloud collected by high-precision laser as P_{gt}. Because the coverage area of GT and the input data differ, we manually cropped the meshes to make their coverage regions as identical as possible. Then, similarly, we uniformly sample points with the same number of P_{gt} on the cropped mesh and use them as the P_{es} for evaluation. In this dataset, we use the resolution of $0.1\,\text{m}$ for marching cubes.

B.2. Dynamic Object Segmentation

Experiment settings. The KTH dataset contains four sequences in total, three of them (KITTI seq. 00, KITTI seq. 05 and Argoverse2) all use 64 beam LiDAR to collect data. We choose $r_{\text{dense}} = 15\,\text{m}$ for the experiments in these sequences. For the Semi-indoor sequence, the sensor is a 16-beam LiDAR, resulting in sparser scans. In this case, we set $r_{\text{dense}} = 8\,\text{m}$ to split the dense and sparse area. Fig. 2 and Fig. 3 demonstrate results of dynamic points removal in KITTI seq. 00 and KITTI seq. 05. Similar to the result depicted in the main text, Erasor [2] and Octomap [3] have a tendency to over-segment dynamic objects, which results in sparser static point clouds shown in zoomed view figures. Additionally, Removert [1] struggles with the complete removal of dynamic objects. Our approach achieves the best performance, with complete removal of dynamic objects while stably preserving static points. On the Semi-indoor sequence, there is an object that remains stationary for a long time, among the methods we tested, only Octomap successfully removed the object in the final map, thus achieving the highest score in this case.

C. Further Implementation Details

As mentioned in the main text, we encode geometric information using only two levels of feature hashing voxels, *i.e.*, $\mathcal{F}^l$, with different resolutions in all our experiments. The highest resolution of the voxel is determined by the scale of the scene. For the *ToyCar3* dataset, it is set to $2\,\text{cm}$, while for other LiDAR-based outdoor datasets, it is set to $30\,\text{cm}$. The resolution of the second voxel level is set to 1.5 times the highest resolution.

For the calculation of L_{eikonal}, we set the perturbation for the numerical gradient calculation ϵ as :

$$\epsilon = \frac{i(\epsilon_{\text{max}} - \epsilon_{\text{min}})}{I}, \tag{6}$$

where, i is the current epoch, I is the total number of epochs, which is 20. For outdoor LiDAR scenes, we set $\epsilon_{\text{max}} = 0.08\,\text{m}$, $\epsilon_{\text{min}} = 0.03\,\text{m}$. However, for *ToyCar3* dataset, as the scale is small, we set $\epsilon_{\text{max}} = 0.8\,\text{cm}$, $\epsilon_{\text{min}} = 0.3\,\text{cm}$.

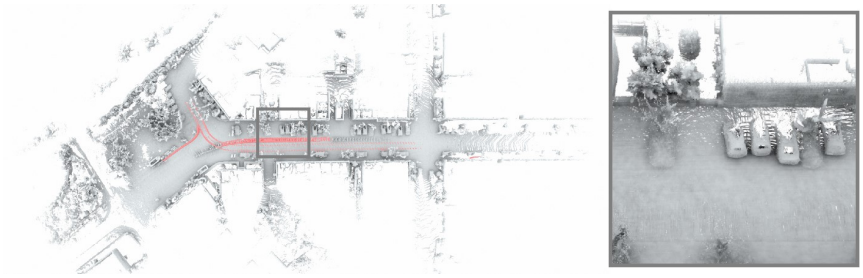
References

[1] Giseop Kim and Ayoung Kim. Remove, Then Revert: Static Point Cloud Map Construction Using Multiresolution Range Images. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020. 2, 3, 4

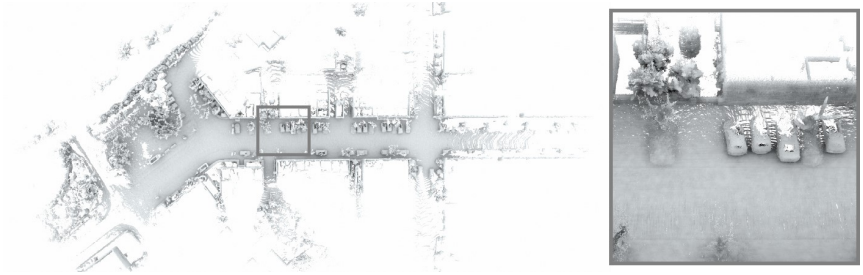
[2] Hyungtae Lim, Sungwon Hwang, and Hyun Myung. ERASOR: Egocentric Ratio of Pseudo Occupancy-Based Dynamic Object Removal for Static 3D Point Cloud Map Building. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2272–2279, 2021. 2, 3, 4

[3] Qingwen Zhang, Daniel Duberg, Ruoyu Geng, Mingkai Jia, Lujia Wang, and Patric Jensfelt. A dynamic points removal benchmark in point cloud maps. In *IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 608–614, 2023. 2, 3, 4

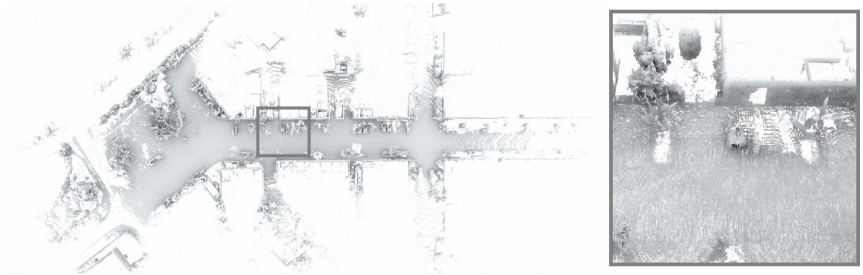
[4] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023. 2



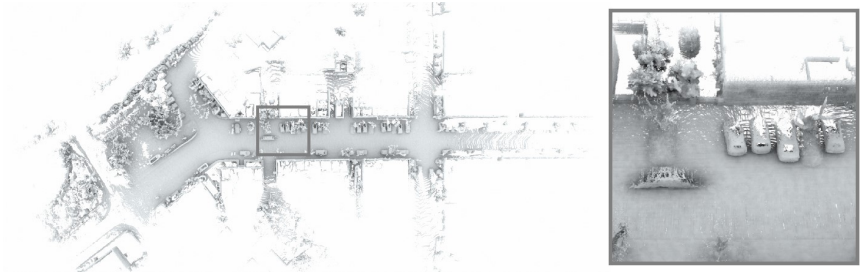
(a) Ground truth



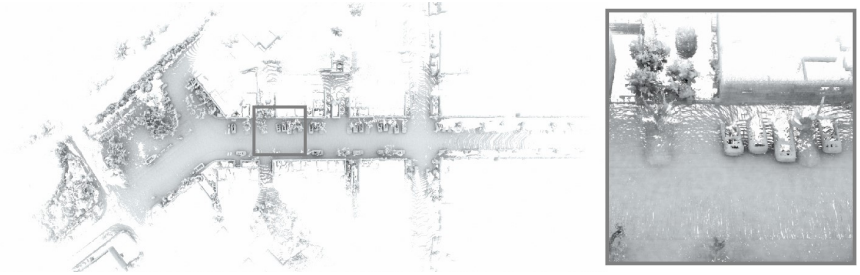
(b) Ours



(c) Erasor [2]

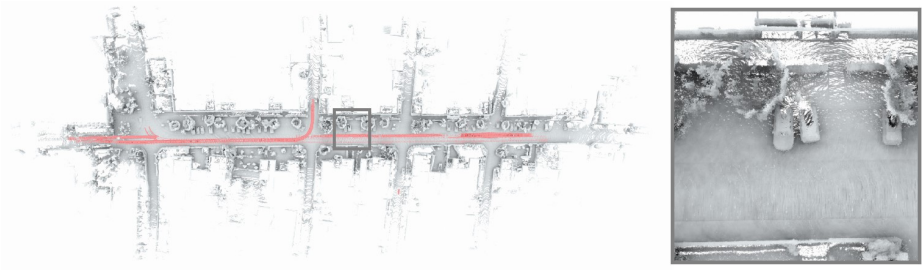


(d) Removert [1]

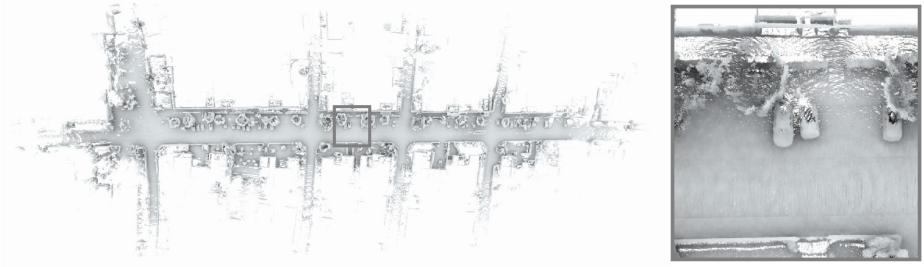


(e) Octomap* [3]

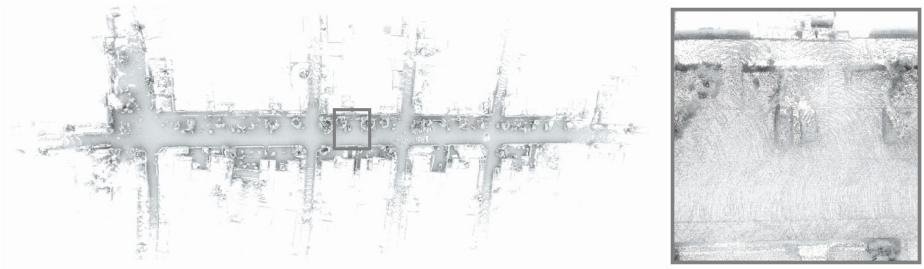
Figure 2. Comparison of dynamic object removal results produced by different methods on the KITTI seq. 00 of the KTH-benchmark. We display the bird's eye view of the complete point cloud with a zoomed view from the gray box. For the ground truth in (a), dynamic objects are marked in red in the bird's eye view, static points are depicted in the zoomed view for clearer comparison.



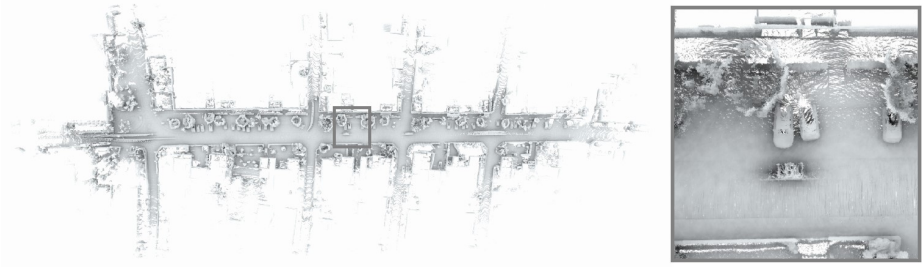
(a) Ground truth



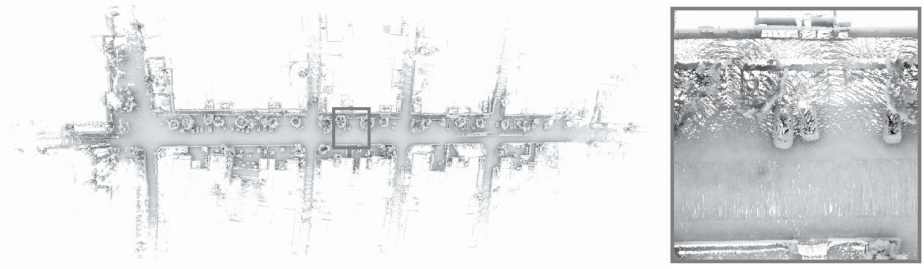
(b) Ours



(c) Erasor [2]



(d) Removert [1]



(e) Octomap* [3]

Figure 3. Comparison of dynamic object removal results produced by different methods on the KITTI seq. 05 of the KTH-benchmark. We display the bird's eye view of the complete point cloud with a zoomed view from the gray box. For the ground truth in (a), dynamic objects are marked in red in the bird's eye view, static points are depicted in the zoomed view for clearer comparison.