Knowledge-Based Systems 284 (2024) 111145

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

City-scale continual neural semantic mapping with three-layer sampling and panoptic representation

Yongliang Shi [a,1], Runyi Yang [b,1], Zirui Wu [c], Pengfei Li [a], Caiyun Liu [a], Hao Zhao [a], Guyue Zhou [a,*]

[a] Institute for AI Industry Research (AIR), Tsinghua University, Beijing 100084, China
[b] Imperial College London, London, SW72AZ, United Kingdom
[c] System Thrust, HKUST(GZ), Guangzhou 511455, China

ARTICLE INFO

Dataset link: https://github.com/liangyongshi/city_siren

Keywords:
Neural mapping
Panoptic representation
Three-layer sampling

ABSTRACT

Neural implicit representations are drawing much attention from the robotics community recently, as they are expressive, continuous and compact. However, city-scale continual implicit dense mapping based on sparse LiDAR input is still an under-explored challenge. To this end, we successfully build a city-scale continual neural mapping system with a panoptic representation that consists of environment-level and instance-level modeling. Given a stream of sparse LiDAR point cloud, it maintains a model that maps 3D coordinates to signed distance field (SDF) values. To address the difficulty of representing geometric information at different levels in city-scale space, we propose a tailored three-layer sampling strategy to sample the global, local and near-surface domains dynamically. Meanwhile, to realize high fidelity mapping of instance under incomplete observation, category-specific prior is introduced to better model the geometric details. We evaluate on the public SemanticKITTI [1] dataset and demonstrate the significance of the newly proposed three-layer sampling strategy and panoptic representation, using both quantitative and qualitative results. Codes and model will be publicly available.

1. Introduction

Mapping is widely recognized as a fundamental environment-sensing capability of intelligent robots [2,3], for example, city-scale 3D maps are critical to the localization and planning of autonomous vehicles. A good mapping method should have **small memory footprint** and **rich map elements** (e.g., geometry and semantics) while allowing **continual updating**. In this paper, we propose a city-scale mapping system that meets these three requirements simultaneously, while addressing several non-trivial technical issues.

With these three requirements in mind, the first question to ask is: Which representation should we use? People have developed various explicit scene representations for mapping such as point clouds [7] (Fig. 1(a)), voxel grids [8], octree map [9], and surfel clouds [10]. Being explicit means that, although these representations differ in detail, they store the coordinates (and other properties) of 3D map points explicitly. As such, they boil down to certain forms of discrete approximation of the underlying 3D map, and their memory footprint inevitably grows with the number of 3D map points.

Contrary to explicit representations, implicitly defined, continuous, differentiable shape representations parameterized by neural network have emerged as a powerful paradigm for surface modeling [11–13]. These methods easily deal with a wide variety of surface topologies with arbitrary resolution, enabling downstream tasks ranging from robotic perception [14] and 3D reconstruction to navigation [15]. Recently, research on RGBD-based continual implicit mappings has made significant progress [16,17]. However, all these studies addressed issues related to indoor scene reconstruction. Major limitations still exist for dense reconstruction in urban scenes from sparse LiDAR utilizing implicit representation, for example: **(1) Scale variation:** Exclusively relying on a uniform global sampling approach is inadequate for fulfilling the demands of reconstructing details [12], especially in urban scenes where the scale varies greatly in different directions, resulting in the difficulty of capturing sufficient local details and making continual update forgetting surface geometry information (Fig. 1(d)). **(2)** The **sparsity and incompleteness** of instance data impairs the effectiveness of reconstruction outcomes, particularly when the instance data is severely deficient due to either scan blind spot or occlusion, which renders complete instance reconstruction arduous (Figs. 1(b) and 1(c)).

In this paper, we propose a continual mapping system with panoptic representation under city-scale scene with LiDAR input (Fig. 1(e)). To

* Corresponding author.
E-mail addresses: shiyongliang@air.tsinghua.edu.cn (Y. Shi), runyi.yang23@imperial.ac.uk (R. Yang), zwu797@connect.hkust-gz.edu.cn (Z. Wu), lipengfei@air.tsinghua.edu.cn (P. Li), liucy.2018@tsinghua.org.cn (C. Liu), zhaohao@air.tsinghua.edu.cn (H. Zhao), zhouguyue@air.tsinghua.edu.cn (G. Zhou).
[1] Equal contribution.

https://doi.org/10.1016/j.knosys.2023.111145
Received 22 August 2023; Received in revised form 30 September 2023; Accepted 30 October 2023
Available online 7 November 2023

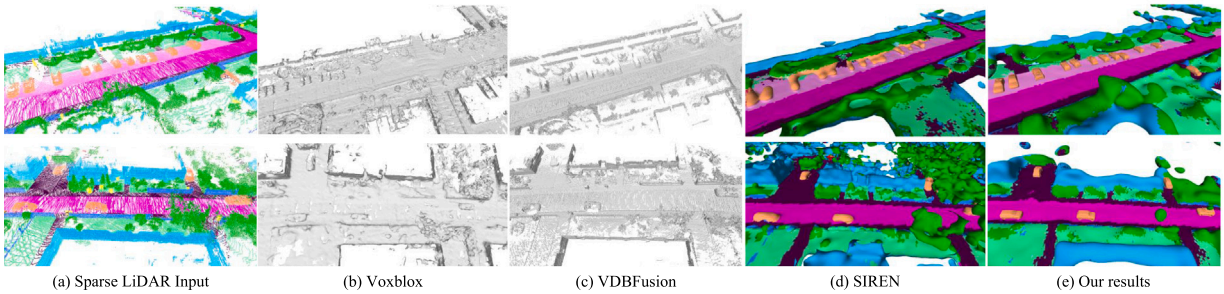

Fig. 1. (a) The input is sparsity-variant point cloud of road scenes captured by LiDAR. Explicit fitting results by (b) Voxblox [4] and (c) VDBFusion [5]. (d) The implicit fitting results of SIREN [6] with our semantic prediction model. (e) Our results.

overcome the memory scalability challenges inherent in traditional explicit reconstruction methods, we employ neural implicit representation. Specifically, we continually update a neural network from sequential LiDAR data, which maps the scene coordinates to signed distance field (SDF) values. This approach allows us to represent the scene while efficiently managing memory usage. Furthermore, we jointly train a parallel neural network for semantic segmentation with the geometry model to better support downstream tasks. To alleviate surface forgetting caused by the **scale variation** of map during continual updates of neural network, our three-layer sampling method ensures that off-surface points from different hierarchical levels, including global, local, and near-surface, all contribute to surface fitting. Additionally, we introduce a category-shared shape prior to generate a prediction of full shape and construct a complete signed distance field of the object, counteracting poor reconstruction quality caused by **sparse** and **incomplete** data of instances in the scene.

To summarize, our contributions are as follows:

- A city-scale continual learning system is developed within a panoptic representation consisting of scene and instance representation.
- Three-layer sampling method is proposed to facilitate the surface fitting of the scene during continual updates.
- An instance representation with category-specific prior is put forward to complete dense reconstruction of incomplete and sparse instances.

2. Related work

2.1. Implicit Neural Scene Representation (INSR)

The ascendancy of neural implicit representations as a formidable paradigm for scene representation has been widely recognized. As a typical approach, Occupancy Networks [18,19] represent the continuous decision boundary of a classifier as an implicit 3D occupancy function, defining a 3D surface. Additionally, the SDF is commonly employed as an implicit representation to capture surface details. However, earlier INSRs based on MLPs with ReLU activations [20,21] struggle with reconstructing high-frequency surface details due to their piecewise linear nature with zero derivatives. Sitzmann introduced MLPs with periodic activation functions for implicit neural representations to address this limitation [6]. Moreover, some studies have focused on enhancing encoding techniques for input coordinates [22]. NeRF [11,23] has garnered significant attention in the domain of large-scale scene reconstruction [24,25] due to their simplicity and exceptional performance.

2.2. Continue learning of scene

Conducting batch training for implicit neural representations becomes impractical when dealing with potentially infinite streams of data. To address this, Yan et al. [17] proposed SDF-based continual neural mapping, updating network parameters upon the arrival of new observations, resulting in a self-improved mapping function. Subsequently, Following the iMAP [26] achieved implicit SLAM in real scenes for the first time, and NICE-SLAM [27] further optimized iMAP with pre-trained geometric priors, enabling detailed reconstruction of large indoor scenes. Meanwhile, iSDF [16] used a neural network to regress input 3D coordinates to signed distance values, facilitating real-time reconstruction from a stream of posed depth images. Azinović [28] incorporated the truncated signed distance function (TSDF) into the NeRF framework to represent surfaces instead of volumetric data. However, all the above are incremental implicit reconstructions of indoor scenes based on RGB-D cameras. Recently, the adoption of an octree-based feature volume for surface representation [13,29] has facilitated significant improvements in reconstructing scene details within the context of SDF-based large-scale incremental 3D mapping.

2.3. Instance representation

The aforementioned works primarily focus on learning representations for entire scenes across a few categories, without delving into the detailed analysis of individual instances. Jiang [30] learn to encode/decode geometric parts of objects at a part scale by training an implicit function auto-encoder, and optimize Latent Implicit Grid representation that matches a partial scene observation. Yang [31] used a shared MLP with instance-specific latent codes to incorporate prior. Kundu [32] uses meta-learning to find a good category-specific initialization and employ instance-specific fully weight encoded functions to represent each object in the scene. Yu et al. [33] introduce the Part-Wise AtlasNet, wherein individual neural networks are assigned the exclusive task of reconstructing specific components of a 3D instance, resulting in improved recovery of 3D objects featuring intricate local structures. Boulch [34] proposed a sampling strategy that selects needles with endpoints on opposite or same sides of the surface to achieve dense reconstruction from sparse point cloud instances. In our contemporaneous work, Ye et al. [35] employed a method similar to that described in this paper for target tracking.

3. Formulation

Fig. 2 depicts the panoptic representation denoted as S, which comprises two subcomponents: ${}^{S}S$ for capturing scene-related information and ${}^{I}S$ for capturing instance-specific details. The former is continually updated using the LiDAR stream, while the latter is constructed based on detected instances and category-specific priors.

Scene: We are committed to modeling 3D environment continually through LiDAR streams D^t with an implicit representation $\mathcal{F}(\cdot)$. The $D^t = (\boldsymbol{p}_i^t, \boldsymbol{s}_i^t)$ is composed of point cloud coordinates $\boldsymbol{p}_i^t$ and corresponding attributes $\boldsymbol{s}_i^t$.

$$\boldsymbol{s}_i^t = \mathcal{F}(\boldsymbol{p}_i^t; \theta^t) \tag{1}$$

Here, t is time stamp, and $\boldsymbol{s}$ is property of scene that is represented by the SDF whose sign indicates whether the region is inside ($-$) or outside

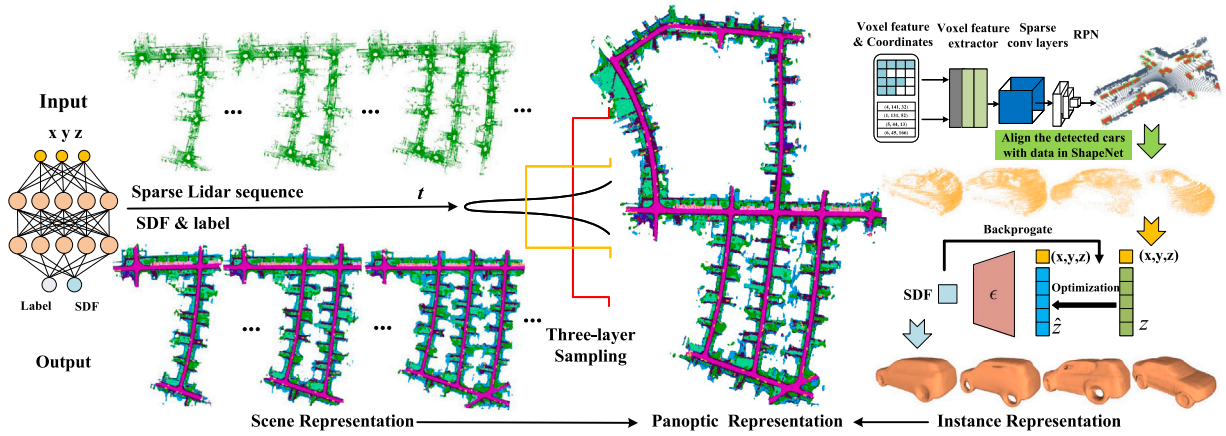

Fig. 2. Given sequential sparse data, our model continuously learns scene property with three-layer sampling strategy that covers different level information including global, local and near-surface to achieve implicit semantic scene representation. In addition, we pre-train a category-specific MLP as prior to complete dense reconstruction of vehicles even with serious data default.

(+) of the shape. $\mathcal{F}(\cdot)$ is a continuous function that maps spatial point $\boldsymbol{p}$ to its distance to the nearest boundary, and the surface ${}^{S}\mathcal{S}$ of the scene is represented by the iso-surface of $\mathcal{F}(\cdot)=0$:

$$ {}^{S}\mathcal{S}=\left\{\boldsymbol{p}_i^t\in\mathbb{R}^3 \mid \mathcal{F}\left(\boldsymbol{p}_i^t;\theta^t\right)=0\right\},\mathcal{F}(\cdot):\mathbb{R}^3\mapsto\mathbb{R}. \tag{2} $$

According to the properties of SDF [16], it is to solve a specific Eikonal boundary value problem which restricts the norm of spatial gradients $\nabla_p\mathcal{F}$ approach to 1 almost everywhere: $\left|\nabla_p\mathcal{F}(\boldsymbol{p})\right|=1$. Moreover, if points are sufficiently close to the surface, the $\mathcal{F}(\boldsymbol{p})$ is expected to be 0, and the gradient $\nabla_p\mathcal{F}$ is equal to the surface normal: $\nabla_p\mathcal{F}(\boldsymbol{p})=\mathbf{n}(\boldsymbol{p})$. Therefore, we fit a neural network $\mathcal{F}(\cdot)$ that parameterize $\boldsymbol{s}$ to model a scene ${}^{S}\mathcal{S}$ from LiDAR streams D^t using a loss of the form:

$$ \mathcal{L}_{\text{sdf}}=\int_{\Omega}\left\|\left|\nabla_p\mathcal{F}(\boldsymbol{p})\right|-1\right\|d\boldsymbol{p}+\int_{\Omega_0}\|\mathcal{F}(\boldsymbol{p})\|+\left(1-\left\langle\nabla_p\mathcal{F}(\boldsymbol{p}),\mathbf{n}(\boldsymbol{p})\right\rangle\right)d\boldsymbol{p} \tag{3} $$

The Ω denotes the entirety of the domain, wherein the zero-level set of the SDF is represented as Ω_0. Additionally, to remedy the lack of constraints on off-surface points, another constraint is introduced:

$$ \mathcal{L}_{\text{off}}=\int_{\Omega\setminus\Omega_0}\psi(\mathcal{F}(\boldsymbol{p}))d\boldsymbol{p} \tag{4} $$

Here, $\psi(\boldsymbol{p})=\exp(-\alpha\cdot|\mathcal{F}(\boldsymbol{p})|)$, $\alpha\gg1$ penalizes off-surface points for creating SDF values close to 0. Generally, we assigns a value of ∞ to all points off the surface, resulting in a consistent contribution for fitting the surface. In the context of mapping large-scale scenes, the effectiveness of **global uniform sampling** may be compromised by the significant scale variations. While global uniform sampling helps to avoid the random prediction of SDF values in free space, the majority of the sampled points are typically invalid with negligible impact on the scene surface during the initial stages of map update. To expedite the fitting of function $\mathcal{F}(\cdot)$, **local sampling** is performed by leveraging information from the current local map. However, with increasing iterations, the neural network may suffer from a loss of pertinent geometric information in the vicinity of the surface. In response, we propose the adoption of a **near-surface sampling** method to address this challenge. As such, the proposed methodology involves sampling off-surface points at three different levels, characterized by sampling proportions λ_g, λ_l and λ_n for global, local, and near-surface sampling, respectively, such that the sum of the proportions equals 1: $\lambda_g+\lambda_l+\lambda_n=1$.

Instance: We denote a shape of instance ${}^{I}\mathcal{S}:\{\boldsymbol{p_1},\boldsymbol{p_2},\ldots,\boldsymbol{p_j}\}$ with a signed distance function f_ϵ:

$$ {}^{I}\mathcal{S}:=\{(\boldsymbol{p_j},s_j):f_\epsilon(\boldsymbol{p_j})=s_j\} \tag{5} $$

Here, $\boldsymbol{p_j}\in\mathbb{R}^3$ are coordinates of the shape. Inspired by DeepSDF [36], an auto-decoder neural network $f_\epsilon(\cdot)$ is trained in a large amount of homogeneous instances with diverse shapes, which contains common properties of this class. Additionally, a latent vector $\boldsymbol{z}$ can be thought of as encoding the desired shape. Given a sparse or partial shape, we will adopt a probabilistic perspective to derive the process of instance reconstruction. The posterior over shape code $\boldsymbol{z}$ which is paired with observed shape ${}^{I}\mathcal{S}$ can be decomposed as:

$$ p\left(\boldsymbol{z}\mid{}^{I}\mathcal{S}\right)=p\left({}^{I}\mathcal{S}\right)\prod_{(\boldsymbol{p_j},s_j)\in{}^{I}\mathcal{S}}p_\epsilon\left(s_j\mid\boldsymbol{z};\boldsymbol{p_j}\right) \tag{6} $$

where ϵ parameterizes SDF prior, and $p_\epsilon\left(s_j\mid\boldsymbol{z};\boldsymbol{p_j}\right)$ is expressed via a deep feed-forward network $f_\epsilon\left(\boldsymbol{z},\boldsymbol{p_j}\right)$:

$$ p_\epsilon\left(s_j\mid\boldsymbol{z};\boldsymbol{p_j}\right)=\exp\left(-\mathcal{L}_{\text{ins}}\left(f_\epsilon\left(\boldsymbol{z},\boldsymbol{p_j}\right),s_j\right)\right) \tag{7} $$

The loss function $\mathcal{L}_{\text{ins}}$ serves to penalize deviations between predicted and actual signed distance function (SDF) values, s_j. Instances denoted by ${}^{I}\mathcal{S}$, where SDF values of points are constrained to zero, is interpreted as likelihoods. During optimization, we maximize the joint log posterior over the reconstructing shape to obtain the shape code $\boldsymbol{z}$:

$$ \hat{\boldsymbol{z}}=\arg\min_{\boldsymbol{z}}\sum_{(\boldsymbol{p_j},s_j)\in{}^{I}\mathcal{S}}\mathcal{L}_{\text{ins}}\left(f_\epsilon\left(\boldsymbol{z},\boldsymbol{p_j}\right),s_j\right)+\frac{1}{\sigma^2}\|\boldsymbol{z}\|_2^2 \tag{8} $$

We assume the latent shape-code space submits to a zero-mean multivariate-Gaussian distribution with a spherical covariance σ^2I. Finally, the $\hat{\boldsymbol{z}}$ are concatenated with relative coordinates, and then feed it into the trained neural network to inference SDF values of reconstructing shape.

4. Method

4.1. Network architecture

Following the network architecture in SIREN [6], We model the SDF s using an MLP with 4 hidden layers of feature size 256, map a 3D coordinate $\boldsymbol{p}=(x,y,z)$ to a SDF value: $\mathcal{F}(\boldsymbol{p};\theta)=s$. Fourier Feature Networks [22] transform the effective neural tangent kernel (NTK) into a stationary kernel with a tunable bandwidth applying Bochner's theorem. We use a Fourier feature mapping to $\gamma(\boldsymbol{p})=[\cos(2\pi\mathbf{B}\boldsymbol{p}),\sin(2\pi\mathbf{B}\boldsymbol{p})]^{\mathrm{T}}$, where each entry in $\mathbf{B}\in\mathbb{R}^{m\times d}$ is sampled from $\mathcal{N}\left(0,\kappa^2\right)$, and κ is chosen for each task and dataset with a hyperparameter sweep. In the absence of any strong prior on the frequency

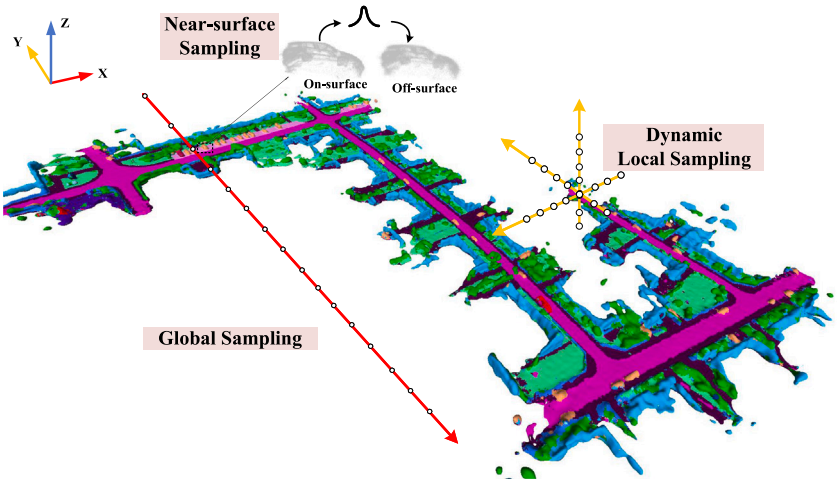

Fig. 3. Three-layer sampling that covers global, local and near-surface.

spectrum of the signal, we use an isotropic Gaussian distribution:

$$\gamma(\boldsymbol{p}) = [a_1 \cos\left(2\pi \mathbf{b}_1^{\mathrm{T}} \boldsymbol{p}\right), a_1 \sin\left(2\pi \mathbf{b}_1^{\mathrm{T}} \boldsymbol{p}\right), \ldots, a_m \cos\left(2\pi \mathbf{b}_m^{\mathrm{T}} \boldsymbol{p}\right), a_m \sin\left(2\pi \mathbf{b}_m^{\mathrm{T}} \boldsymbol{p}\right)]^{\mathrm{T}} \tag{9}$$

In addition, we add a network with the same structure in parallel to output the semantic value of each point. The architectural framework utilized for the purpose of instance reconstruction is based on the DeepSDF [36] paradigm.

4.2. Sampling

As delineated in Section 3, the $\mathcal{L}_{\mathrm{off}}$ can be cast in loss functions to penalize deviations, where the off-surface point constraints are represented by $\psi(\mathcal{F}(\boldsymbol{p}))$. To accommodate new data streams, we adopt a practical approach by sampling D^t over the entire domain Ω, which includes both on-surface points Ω_0 whose SDF values are 0, as well as off-surface points $\Omega \backslash \Omega_0$, with SDF values set to ∞.

On-surface points sampling: To balance point cloud density and computing efficiency, we select keyframes at regular intervals of three frames to update the neural network. To mitigate catastrophic forgetting, 75% of points are randomly sampled from previous keyframes and 25% from the latest ones. However, redundant samples of buildings and roads, along with decreasing representation of small instances as the scene size increases, may lead to inadequate surface fitting. To overcome this, we employ importance sampling based on semantic information, extracting N_g on-surface points within a vicinity of n_o points of instances at each iteration. For this study, N_g and n_o are set to 140,000 and 6000, respectively.

Three-layer Sampling for Off-surface points: The present study concerns a key challenge in global uniform sampling, where the majority of points fall in free space, causing a loss of surface information due to uniform penalty assignment. Specifically, as the SDF values are uniformly set to ∞ for all points off the surface, points near the surface receive the same penalty as points far away from the surface. To overcome this limitation, a novel three-layer sampling strategy has been proposed, as illustrated in Fig. 3. This strategy leverages the complementary strengths of different sampling methods by incorporating three levels of information, namely global, local, and near-surface sampling, each with a different proportion λ_g, λ_l, and λ_n.

For a limited scene or an individual object, the point cloud's scale in all directions remains close, and off-surface points obtained through global uniform sampling are adequate to represent different levels of noise. However, in an urban scene where the map continually updates, the scale difference in all directions gradually intensifies. Consequently, the scale map in the **Z** direction becomes almost negligible compared to the scale of the scene in the **X** and **Y** directions. Hence, uniform sampling of the entire space alone is insufficient to represent the noise of local geometry. The results depicted in Fig. 8 demonstrate that the network trained exclusively on global sampling mainly learned the overall outline information of the scene. Nevertheless, the overall contour information remains vital, and thus a certain proportion of points, denoted by $\boldsymbol{p_g}$, acquired through global sampling is essential:

$$\begin{aligned} \boldsymbol{p_g} &= \mathcal{U}([-1,-1,-1],[1,1,1]), \\ \lambda_g &= size(\boldsymbol{p_g})/size(N_g). \end{aligned} \tag{10}$$

If the continual update employs global uniform sampling, it may result in negligible contributions from most off-surface points at the onset, leading to the loss of local geometric details. In response to the influx of new LiDAR stream, the dynamic boundary ($\boldsymbol{b_l}$ and $\boldsymbol{b_u}$) of the scene is estimated, where $\boldsymbol{b_l}$ denotes the lower limit and $\boldsymbol{b_u}$ denotes the upper limit of the scene boundary. Subsequently, local off-surface points $\boldsymbol{p_l}$ are uniformly sampled within this range to maximize the network's ability to capture the local scene changes, $\boldsymbol{p_l} \sim \mathcal{U}(\boldsymbol{b_l}, \boldsymbol{b_u})$.

$$\begin{aligned} \boldsymbol{b_l} &= \left(\frac{\boldsymbol{L_{min}} - G_{min}}{G_{max} - G_{min}} - 0.5\right) \times 2 \\ \boldsymbol{b_u} &= \left(\frac{\boldsymbol{L_{max}} - G_{min}}{G_{max} - G_{min}} - 0.5\right) \times 2 \end{aligned} \tag{11}$$

Here, the maximum and minimum coordinate vectors of the extant local point clouds are denoted by $\boldsymbol{L_{max}}(x^l_{max}, y^l_{max}, z^l_{max})$ and $\boldsymbol{L_{min}}(x^l_{min}, y^l_{min}, z^l_{min})$, respectively, both of which adjust dynamically to accommodate the influx of new data. G_{max} and G_{min} are the maximum and minimum coordinate values of the entire scene. However, local dynamic sampling in isolation is inadequate, owing to the fact that off-surface points lying outside the local scene do not contribute to the network, which makes the neural network predict random SDF values in free space. Therefore, a certain proportion of local and global sampling are both necessary.

The aforementioned sampling strategies provide a comprehensive and adaptable depiction of the entire and local scene. However, as the volume of data increases, the effects of local sampling become increasingly similar to those of global sampling, with the exception of the Z dimension's spatial scale. Consequently, the neural network gradually loses sight of the surface information of the scene, which is illustrated in Fig. 4(a). As a result, near-surface points $\boldsymbol{p_n}$ are sampled to aggravate the penalty of noise near the surface in the learning process:

$$\boldsymbol{p_n} = \left\{(\boldsymbol{p} + \mathbf{h}, \boldsymbol{p} - \mathbf{h}) \mid \boldsymbol{p} \in S, \mathbf{h} \sim \mathcal{N}\left(0, \sigma_h\right)\right\} \tag{12}$$

where $\mathbf{h}$ is randomly sampled from the multivariate Gaussian distribution $\mathcal{N}\left(0, \sigma_h\right) \in \mathbb{R}^3$ with standard deviation σ_h, σ_h is $diag(0.0003)$ in this paper.

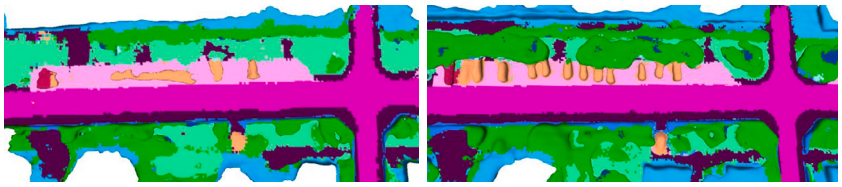

Fig. 4. (a) Without near-surface sampling, the network gradually forgets the information of instances like cars and trees in the scene. (b) Near-surface sampling successfully mitigates catastrophic forgetting of instances.

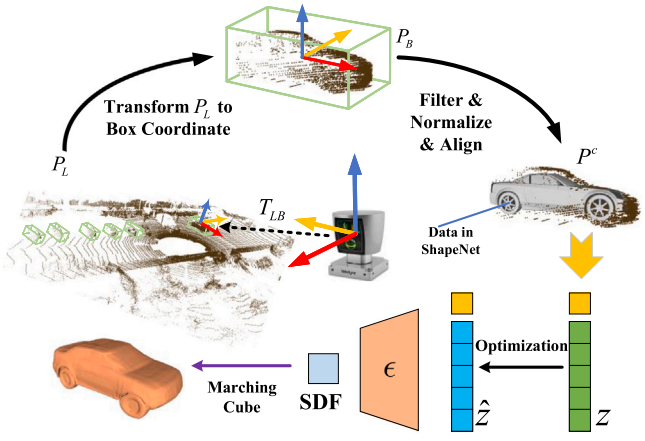

Fig. 5. Instance reconstruction with category-specific shape prior [37].

4.3. Instance representation

In the scene reconstruction, the outcomes of the instances exhibit a deficiency of intricate particulars. As an illustration, vehicles appear as convex components on the map, as depicted in Fig. 4(b). One of the core benefits of an object-aware approach is able to incorporate inductive bias that objects instances within the same category often have similar 3D shapes and appearance. Taking this as inspiration, we introduce a category-specific prior by sharing a neural network across the object instances to well depict details of instances with absent data in the map. Taking the car as an example, we first use SECOND [38] to detect the bounding box of cars and transfer the corresponding point cloud from the LiDAR coordinate system to the central coordinate system of the bounding box, $P_B = {T_{LB}}^{-1} \cdot P_L$, where T_{LB} is the transformation of the bounding box center with respect to the LiDAR coordinate system, P_L is the raw LiDAR data, P_B is the coordinate relative to bounding box center coordinate system. We utilize a pass-through filter to extract the vehicle point cloud P^c bounded by the bounding box from P_B. Next, normalize P^c to [−0.5,0.5] with the range of bounding box as the maximum and minimum value and align with the data in ShapeNet [37]. What is next, given fixing ϵ trained in cars of ShapeNet by DeepSDF [36], a latent code $\hat{\boldsymbol{z}}_i$ for vehicle P_i^c can be estimated via MAP estimation. We concatenate the $\hat{\boldsymbol{z}}_i$ with the relative coordinates of the vehicle, then feed them into the MLP, as Fig. 5 depicts. Adam optimizer is used to update the latent code, and complete the reconstruction of the vehicle instance via Marching Cube [39] according to the SDF value predicted by the network. Finally, the vehicle is scaled and converted to the scene map coordinate system according to the pose of the center relative to the bounding box, the panoptic scene is represented in Fig. 1(e).

4.4. Training and inference

Training: For scene representation, besides the geometry constraints described in Section 3, we also add a parallel implicit generative head to directly model the implicit semantic field. Its structure is similar to our SDF model, except that it outputs the probabilities of label classification. We supervise the semantic segmentation results with a multi-classification cross-entropy loss:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{N_g}\sum_{i=1}^{N_g}\sum_{c=1}^{C} y_{i,c}\log(pr_{i,c}). \tag{13}$$

where $y_{i,c}$ and $pr_{i,c}$ are the actual and predicted probability for point i belonging to category c respectively. As per the prescribed constraints, N_g points are sampled at random from Ω_0, while an equal number of off-surface points are selected from $\Omega \backslash \Omega_0$, employing our three-layer sampling strategy that involves the selection of λ_g, λ_l and λ_n in the proportions of 0.55, 0.35 and 0.1, respectively. The optimization of the scene representation is executed by minimizing the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{sdf}} + \mathcal{L}_{\text{off}} + \mathcal{L}_{\text{seg}} \tag{14}$$

Regarding instance representation, given a set of cars of ShapeNet, we train a category-shared MLP f_ϵ, following the DeepSDF approach, whereby we minimize the aggregate of losses between the predicted and actual SDF values of the points within the cars, subject to the following loss function:

$$\mathcal{L}_{\text{ins}}\left(f_\epsilon(\boldsymbol{p}), s\right) = \left|\text{clamp}\left(f_\epsilon(\boldsymbol{p}), \delta\right) - \text{clamp}(s, \delta)\right|, \tag{15}$$

where $\text{clamp}(\boldsymbol{p}, \delta) := \min(\delta, \max(-\delta, \boldsymbol{p}))$ introduces the parameter δ to control the distance from the surface over which we expect to maintain a metric SDF.

Inference: Given the SDF values and semantic labels by our scene representation, semantic mapping is performed. For city scene, scale in Z direction is almost negligible comparing with the scales in X and Y. When inference, sampling same number of points in all directions like the Marching Cube will lead to large amount of invalid samples in Z-axis, which gives rise to insufficiency of memory usage. In view of this, we will sample $N_x \times N_y \times N_z$ points, where $N_x = N_y$, and the N_z is:

$$N_z = \frac{Z_{max} - Z_{min}}{G_{max} - G_{min}} \times N_x. \tag{16}$$

Here, Z_{max} and Z_{min} are the maximum and minimum values of the map on the Z-axis respectively, G_{max} and G_{min} are the maximum and minimum values of the map in all directions. We sample points on the Z-axis from the position Z_{start} where there is information instead of 0, and the Z_{start} is:

$$Z_{start} = \frac{Z_{min} - G_{min}}{G_{max} - G_{min}} \times N_x. \tag{17}$$

As a consequence, we can guarantee optimal utilization of memory resources while generating high-resolution meshes. Simultaneously, we will render the produced mesh in color, with the color of each vertex determined by the label of its closest spatial point. The outcomes depicted in Fig. 6 demonstrate a discernible superiority of our proposed approach in comparison to the conventional methodology.

For the representation of instance, given a normalized car detected from LiDAR sequence, we firstly feed the concatenated vector including coordinates and initialized latent code to the f_ϵ, and optimize the latent

Fig. 6. (a) Result of Marching Cube. (b) Our method is capable of outputting more details under the same memory utilization.

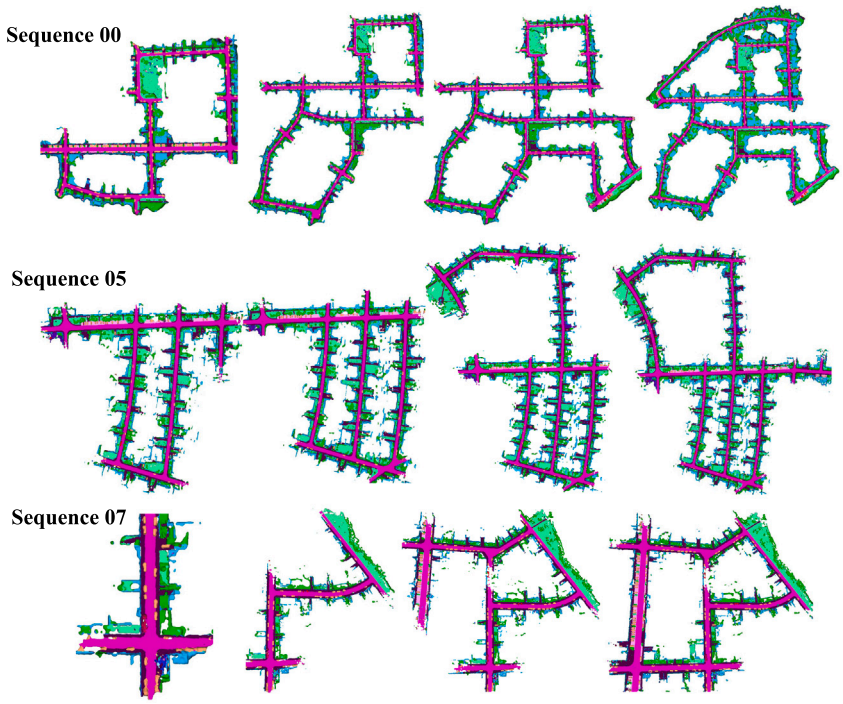

Fig. 7. We select three LiDAR odometry sequences of different sizes from the SemanticKITTI: small (07), medium (05), and large (00) to execute continual implicit semantic mapping.

code through back-propagation. The coordinates and corresponding latent code with fixed parameters are cascaded into the trained neural network, the SDF values are output and the Marching Cube [39] is used to generate the corresponding mesh.

5. Experiment and results

We evaluate our method on three city-scale sequences of SemanticKITTI [1] odometry at large, medium, and small magnitudes: 00 (4541 scans), 05 (2761 scans) and 07 (1101 scans), and visualization results are shown in Fig. 7. All experiments are conducted on a Linux system with Intel Core i9-12900K CPU at 5.2 GHz, and NVIDIA GeForce RTX A4000 GPU with 16 GB of memory.

5.1. Data and metrics

Data Preparation: Before training, the outliers and dynamic information are deleted in line with the ground truth of the semantic label. In addition, to obtain the prior for our cars, we trained the category-specific shared MLP on cars of ShapeNet [37]. The reconstructing cars of every frame are extracted and normalized to [−0.5, 0.5], as described in section IV(C). Owing to a slight deviation in the pose of data obtained by the 3D detection network, we fine-tune the angle to align our cars as closely as possible with the car in ShapeNet [37].

Metrics: We evaluate both the reconstruction quality and semantic segmentation of the system. For reconstruction quality evaluation, we uniformly sample 1,000,000 points from the ground-truth points and reconstructed meshes, respectively, and then report the following metrics. *Chamfer Distance* (denoted as CD in Tables 1 2 4) finds the nearest point in the other point set, and sums the square of distance up. *Precision* is the fraction of the points from the reconstructed mesh that is closer to points in the ground truth than a threshold distance, which is set to 0.5 m. *Recall* refers to the fraction of the ground truth points closer to the points in the reconstruction mesh than 0.5 m. *Fscore* is the average of accuracy and completeness and is used to quantify the overall reconstruction quality. Besides, the *mean interactions over union (mIoU)* are used in semantic segmentation evaluation. In the tables, bold and black font represents the optimal choice, while red font indicates the second-best option.

5.2. Ablation study for three-layer sampling method

Sampling strategy and iteration number play important roles in implicit reconstruction. The reconstruction results in Fig. 8 show that: With the increasing of iteration, adopting only global uniform sampling can easily lead to the forgetting of local geometry, meanwhile, the neural network learns some floating noise into the scene as well; On the basis, within global and local sampling, both reconstruction and

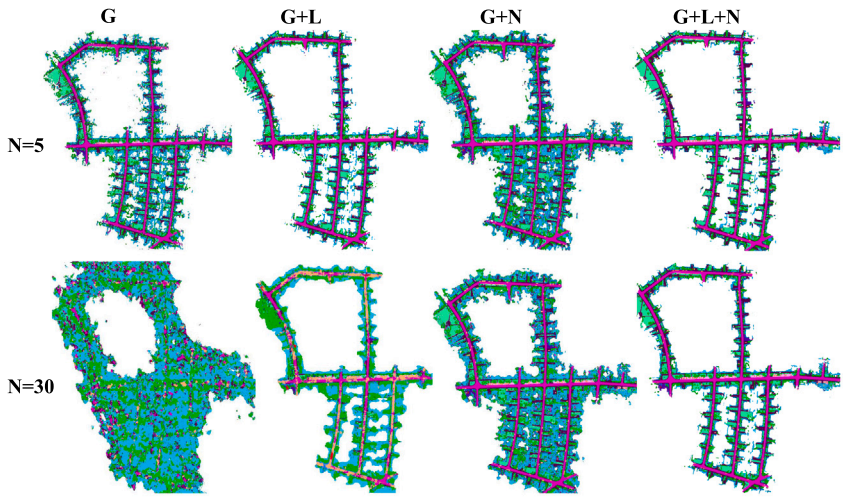

Fig. 8. Ablation study for sampling strategy with a different number of iterations. We denote N as the iteration number, G, L and N represent the global, local and near-surface sampling, respectively.

Table 1
Ablation study for Three-layer Sampling strategy.

Iteration (epochs)	Global sampling	Local sampling	Near-surface sampling	CD (m)	mIoU (%)
5	✓	×	×	0.25	67.6
	✓	✓	×	2.12	95.0
	✓	×	✓	0.65	95.2
	✓	✓	✓	**0.09**	**96.6**
30	✓	×	×	3.33	72.4
	✓	✓	×	4.55	74.9
	✓	×	✓	1.00	96.1
	✓	✓	✓	**0.05**	**96.3**

Table 2
Evaluation on reconstruction quality and segmentation of different encoding methods when the iteration number is 5.

Encoding methods	CD (m)	mIoU (%)
Fourier	**0.09**	**96.6**
Positional	1.05	77.8
L-Fourier	7.17	90.0
No encoding	7.12	76.0

semantic segmentation are improved in visualization, which alleviate the isolated point noise; After utilizing global and near-surface sampling, significant improvements in both reconstruction quality and semantic segmentation were observed in visualization and quantitative results. However, there was an issue of excessive local patch completion. Having employed the three-layer sampling method, the reconstruction and semantic segmentation achieve the best result. Quantitative evaluation results in Table 1 prove the effectiveness and necessity of the three-layer sampling method, which is robust to changes of iterations during training.

Encoding methods: For coordinate-based MLPs, passing input points through a encoding method on a regression task is a prevailing practice. We compare the performance of our task with no input encoding and three encoding methods. One is Positional encoding that is consistent with the work proposed by Rahaman [20] and its encoding level is 10, the other is Fourier encoding [22] with an isotropic Gaussian distribution used in this paper, and the last is Learnable Fourier (L-Fourier) encoding [40] that is the state-of-art method, as shown in Fig. 9. Table 2 shows that, in the case of using the three-layer sampling method, Fourier encoding emerges as the most efficacious in tasks pertaining to semantic segmentation and reconstruction. This preeminence can be attributed to the inherent capability of Fourier encoding to encompass a more extensive spectrum of spatial interrelationships and intricate patterns, which surpasses the capabilities of Positional encoding. Conversely, the Learnable Fourier encoding, due to its expanded parameter space, encounters challenges in achieving the desired state of convergence within a constrained ambit of iterative steps.

5.3. Reconstruction quality of instance representation

We present a comparison of our method against two direct (non-learned) methods, Poisson meshing [41] and Ball Pivoting [42], and a learning-based method SIREN [6]. The instances we will evaluate are obtained by the 3D detection algorithm without the complementary ground truth, so the reconstruction quality is only analyzed in terms of visualization (Fig. 10) and is not quantitatively evaluated. As expected, the direct methods fail to fit a proper shape let alone predict the missing part. While Poisson meshing barely contains any detail, Ball Pivoting however produces a detailed mesh around the input point cloud but it fails to reconstruct the hidden parts. For SIREN, once there are pieces of missing data, it is easy to cause underfitting and fail to reconstruct a complete instance. In light of the precedent established by category-specific priors, our method exhibits the capacity to prognosticate values pertaining to the absent components. This, in turn, facilitates the attainment of a highly authentic reconstruction for instances characterized by sparsity and incompleteness.

5.4. Comparative analysis of reconstruction methods

We conducted a comparative analysis of various reconstruction algorithms, encompassing two explicit methods, namely Voxblox [4] and VDBfusion [5], alongside an implicit representation, denoted as SIREN [6]. As delineated in Table 3, our proposed model is comprised of a scene model (1.8MB) in conjunction with a category-specific prior (2.4 MB). Notably, the memory requirement of our model does not escalate with the expansion of the scene's size, a characteristic that it shares with SIREN, which has the least model memory requirement (1.8 MB). In stark contrast, explicit representations not only demand a pronounced memory overhead but also frequently manifest conspicuous discrepancies and omissions in their reconstruction outcomes.

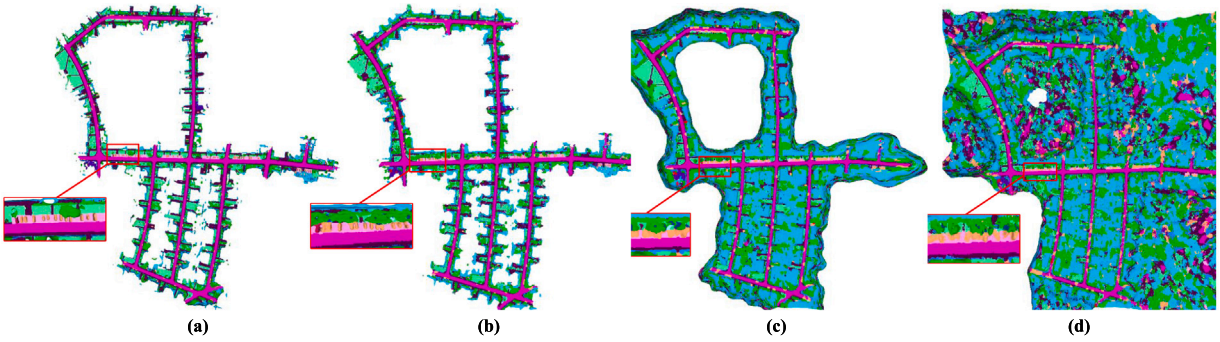

Fig. 9. Representations with (a) Fourier encoding. (b) Positional encoding. (c) Learnable Fourier encoding. (d) no encoding.

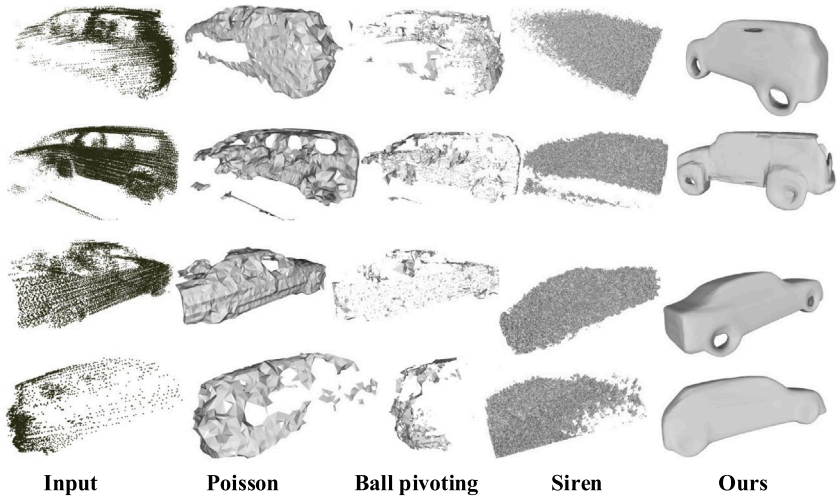

Fig. 10. Qualitative comparison of our instance representation to other reconstruction methods in KITTI.

Table 3
Memory of different methods (MB).

Sequence	Voxblox	VDBfusion	SIREN	Ours
00	448.9	3012.2	**1.8**	4.2
05	264.6	2103.3	**1.8**	4.2
07	82.4	748.1	**1.8**	4.2

Table 4
Reconstruction Quality Evaluation on KITTI when the threshold is 0.5 m.

Methods	CD (m)	Precision (%)	Recall (%)	Fscore (%)
Voxblox	**0.02**	**99.87**	97.33	**98.58**
VDBFusion	**0.02**	92.64	**98.19**	95.34
SIREN	1.05	21.57	6.55	10.05
Ours	0.05	96.81	70.90	81.86

This renders them unsuitable for efficiently processing expansive environments. While our method may marginally lag behind conventional techniques in the context of scene reconstruction quality assessment (Table 4), it exhibits a distinct edge in addressing data deficiencies stemming from occlusions and constraints in sensor resolution.

6. Conclusions

We have introduced an expansive, city-scale continual learning framework encompassing a comprehensive panoptic representation that spans both scene and instance levels. Concerning scene-level dynamics, the integration of new LiDAR streams triggers the application of a three-layer sampling strategy, strategically devised to facilitate the assimilation of global, local, and surface-proximate information. At the instance level, a category-shared prior is pre-trained to serve as a foundational element for the implicit reconstruction of individual instances. In comparison to explicit representations [4,5] and implicit representation [6], our proposed approach exhibits an advantageous equilibrium between the parsimonious utilization of memory resources and the attainment of reconstructions distinguished by high quality. In the future, we aspire to enhance the fidelity of our reconstruction outputs through the integration of additional prior information. Firstly, we plan to introduce dense visual geometric priors [43] akin to MonoSDF [44], as photoconsistency cues are conducive to the establishment of globally accurate 3D geometry in textured regions, whereas normal and depth cues merely furnish local geometric information. Concurrently, we possess the capability to predict RGB for our map, thus generating a hybrid implicit field of NeRF + SDF for city-scale scenes. Secondly, by introducing pre-trained 3D implicit neural assets [45], our approach is primed to reconstruct a more diversified spectrum of colorful instance 3D reconstructions. Lastly, through the introduction of a scene-level geometric prior generative model [46], the reconstruction of architecture and background (including streets, green belts, and so on) is rendered more realistic.

CRediT authorship contribution statement

Yongliang Shi: Resources, Project administration, Methodology, Investigation, Formal analysis. **Runyi Yang:** Visualization, Software, Project administration, Data curation. **Zirui Wu:** Writing – original draft, Software. **Pengfei Li:** Software, Formal analysis. **Caiyun Liu:**

Data curation. **Hao Zhao:** Writing – original draft, Methodology, Conceptualization. **Guyue Zhou:** Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://github.com/liangyongshi/city_siren.

Acknowledgments

This work was supported by the Tsinghua-Toyota Joint Research Fund, China (20223930097).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.knosys.2023.111145.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall, Semantickitti: A dataset for semantic scene understanding of lidar sequences, in: 2019 IEEE/CVF International Conference on Computer Vision, (ICCV), 2019, pp. 9296–9306, http://dx.doi.org/10.1109/ICCV.2019.00939.

[2] J.-R. Ruiz-Sarmiento, C. Galindo, J. Gonzalez-Jimenez, Building multiversal semantic maps for mobile robot operation, Knowl.-Based Syst. 119 (2017) 257–272, http://dx.doi.org/10.1016/j.knosys.2016.12.016, URL https://www.sciencedirect.com/science/article/pii/S0950705116305184.

[3] S. Pu, G. Vosselman, Knowledge based reconstruction of building models from terrestrial laser scanning data, ISPRS J. Photogramm. Remote Sens. 64 (6) (2009) 575–584, http://dx.doi.org/10.1016/j.isprsjprs.2009.04.001, URL https://www.sciencedirect.com/science/article/pii/S0924271609000501.

[4] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, J. Nieto, Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), 2017, pp. 1366–1373, http://dx.doi.org/10.1109/IROS.2017.8202315.

[5] I. Vizzo, T. Guadagnino, J. Behley, C. Stachniss, VDBFusion: Flexible and efficient TSDF integration of range sensor data, Sensors 22 (3) (2022) http://dx.doi.org/10.3390/s22031296, URL https://www.mdpi.com/1424-8220/22/3/1296.

[6] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 7462–7473, URL https://proceedings.neurips.cc/paper_files/paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf.

[7] D. Rozenberszki, A.L. Majdik, LOL: Lidar-only odometry and localization in 3D point cloud maps, in: 2020 IEEE International Conference on Robotics and Automation, (ICRA), 2020, pp. 4379–4385, http://dx.doi.org/10.1109/ICRA40945.2020.9197450.

[8] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W.T. Freeman, J.B. Tenenbaum, Learning shape priors for single-view 3D completion and reconstruction, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 673–691, http://dx.doi.org/10.1007/978-3-030-01252-6_40.

[9] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots 34 (2013) 189–206, URL https://api.semanticscholar.org/CorpusID:8655888.

[10] J. Behley, C. Stachniss, Efficient surfel-based SLAM using 3D laser range data in urban environments., in: Robotics: Science and Systems, Vol. 2018, 2018, p. 59, http://dx.doi.org/10.15607/RSS.2018.XIV.016.

[11] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, in: European Conference on Computer Vision, Springer, 2020, pp. 405–421, http://dx.doi.org/10.1007/978-3-030-58452-8_24.

[12] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, S. Fidler, Neural geometric level of detail: Real-time rendering with implicit 3D shapes, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2021, pp. 11353–11362, http://dx.doi.org/10.1109/CVPR46437.2021.01120.

[13] X. Zhong, Y. Pan, J. Behley, C. Stachniss, SHINE-mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations, in: 2023 IEEE International Conference on Robotics and Automation, (ICRA), 2023, pp. 8371–8377, http://dx.doi.org/10.1109/ICRA48891.2023.10160907.

[14] D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, M. Hutter, Neural scene representation for locomotion on structured terrain, IEEE Robot. Autom. Lett. 7 (4) (2022) 8667–8674, http://dx.doi.org/10.1109/LRA.2022.3184779.

[15] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, M. Schwager, Vision-only robot navigation in a neural radiance world, IEEE Robot. Autom. Lett. 7 (2) (2022) 4606–4613, http://dx.doi.org/10.1109/LRA.2022.3150497.

[16] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, M. Mukadam, iSDF: Real-time neural signed distance fields for robot perception, in: Robotics: Science and Systems, 2022, http://dx.doi.org/10.48550/arXiv.2204.02296.

[17] Z. Yan, Y. Tian, X. Shi, P. Guo, P. Wang, H. Zha, Continual neural mapping: Learning an implicit scene representation from sequential observations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15782–15792, http://dx.doi.org/10.1109/ICCV48922.2021.01549.

[18] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, A. Geiger, Convolutional occupancy networks, in: European Conference on Computer Vision, Springer, 2020, pp. 523–540, http://dx.doi.org/10.1007/978-3-030-58580-8_31.

[19] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger, Occupancy networks: Learning 3D reconstruction in function space, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 4455–4465, http://dx.doi.org/10.1109/CVPR.2019.00459.

[20] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, A. Courville, On the spectral bias of neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 5301–5310, URL https://proceedings.mlr.press/v97/rahaman19a.html.

[21] Z.-Q.J. Xu, Y. Zhang, Y. Xiao, Training behavior of deep neural network in frequency domain, in: International Conference on Neural Information Processing, Springer, 2019, pp. 264–274, http://dx.doi.org/10.1007/978-3-030-36708-4_22.

[22] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 7537–7547, URL https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf.

[23] Z. Gai, Z. Liu, M. Tan, J. Ding, J. Yu, M. Tong, J. Yuan, EGRA-NeRF: Edge-guided ray allocation for neural radiance fields, Image Vis. Comput. 134 (2023) 104670, http://dx.doi.org/10.1016/j.imavis.2023.104670, URL https://www.sciencedirect.com/science/article/pii/S0262885623000446.

[24] A. Liu, S. Zhang, C. Zhang, S. Zhi, X. Li, RaNeRF: Neural 3-D reconstruction of space targets from ISAR image sequences, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–15, http://dx.doi.org/10.1109/TGRS.2023.3298067.

[25] Y. Wei, S. Liu, J. Zhou, J. Lu, Depth-guided optimization of neural radiance fields for indoor multi-view stereo, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 10835–10849, http://dx.doi.org/10.1109/TPAMI.2023.3263464.

[26] E. Sucar, S. Liu, J. Ortiz, A.J. Davison, iMAP: Implicit mapping and positioning in real-time, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6229–6238, http://dx.doi.org/10.1109/ICCV48922.2021.00617.

[27] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M.R. Oswald, M. Pollefeys, NICE-slam: Neural implicit scalable encoding for SLAM, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2022, pp. 12776–12786, http://dx.doi.org/10.1109/CVPR52688.2022.01245.

[28] D. Azinović, R. Martin-Brualla, D.B. Goldman, M. Nießner, J. Thies, Neural RGB-d surface reconstruction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6290–6301, http://dx.doi.org/10.1109/CVPR52688.2022.00619.

[29] X. Yu, Y. Liu, S. Mao, S. Zhou, R. Xiong, Y. Liao, Y. Wang, NF-atlas: Multi-volume neural feature fields for large scale lidar mapping, IEEE Robot. Autom. Lett. 8 (9) (2023) 5870–5877, http://dx.doi.org/10.1109/LRA.2023.3300281.

[30] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser, Local implicit grid representations for 3D scenes, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2020, pp. 6000–6009, http://dx.doi.org/10.1109/CVPR42600.2020.00604.

[31] B. Yang, Y. Zhang, Y. Xu, Y. Li, H. Zhou, H. Bao, G. Zhang, Z. Cui, Learning object-compositional neural radiance field for editable scene rendering, in: 2021 IEEE/CVF International Conference on Computer Vision, (ICCV), 2021, pp. 13759–13768, http://dx.doi.org/10.1109/ICCV48922.2021.01352.

[32] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. Guibas, A. Tagliasacchi, F. Dellaert, T. Funkhouser, Panoptic neural fields: A semantic object-aware neural scene representation, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2022, pp. 12861–12871, http://dx.doi.org/10.1109/CVPR52688.2022.01253.

[33] Q. Yu, C. Yang, H. Wei, Part-wise AtlasNet for 3D point cloud reconstruction from a single image, Knowl.-Based Syst. 242 (2022) 108395, http://dx.doi.org/10.1016/j.knosys.2022.108395, URL https://www.sciencedirect.com/science/article/pii/S0950705122001587.

[34] A. Boulch, P.-A. Langlois, G. Puy, R. Marlet, NeeDrop: Self-supervised shape representation from sparse point clouds using needle dropping, in: 2021 International Conference on 3D Vision, (3DV), 2021, pp. 940–950, http://dx.doi.org/10.1109/3DV53792.2021.00102.

[35] J. Ye, Y. Chen, N. Wang, X. Wang, Online adaptation for implicit object tracking and shape reconstruction in the wild, IEEE Robot. Autom. Lett. 7 (4) (2022) 8909–8916, http://dx.doi.org/10.1109/LRA.2022.3189185.

[36] J.J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, Deepsdf: Learning continuous signed distance functions for shape representation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2019, pp. 165–174, http://dx.doi.org/10.1109/CVPR.2019.00025.

[37] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., Shapenet: An information-rich 3d model repository, 2015, http://dx.doi.org/10.48550/arXiv.1512.03012, arXiv preprint arXiv:1512.03012.

[38] Y. Yan, Y. Mao, B. Li, SECOND: Sparsely embedded convolutional detection, Sensors 18 (10) (2018) http://dx.doi.org/10.3390/s18103337, URL https://www.mdpi.com/1424-8220/18/10/3337.

[39] W.E. Lorensen, History of the marching cubes algorithm, IEEE Comput. Graph. Appl. 40 (2) (2020) 8–15, http://dx.doi.org/10.1109/MCG.2020.2971284.

[40] Y. Li, S. Si, G. Li, C.-J. Hsieh, S. Bengio, Learnable Fourier features for multi-dimensional spatial positional encoding, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 15816–15829, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/84c2d4860a0fc27bcf854c444fb8b400-Paper.pdf.

[41] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson Surface Reconstruction, in: A. Sheffer, K. Polthier (Eds.), Symposium on Geometry Processing, The Eurographics Association, 2006, http://dx.doi.org/10.2312/SGP/SGP06/061-070.

[42] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, IEEE Trans. Vis. Comput. Graphics 5 (4) (1999) 349–359, http://dx.doi.org/10.1109/2945.817351.

[43] A. Eftekhar, A. Sax, J. Malik, A. Zamir, Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3D scans, in: 2021 IEEE/CVF International Conference on Computer Vision, (ICCV), 2021, pp. 10766–10776, http://dx.doi.org/10.1109/ICCV48922.2021.01061.

[44] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, A. Geiger, Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction, in: Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 25018–25032, URL https://niujinshuchong.github.io/monosdf/.

[45] B. Shen, X. Yan, C.R. Qi, M. Najibi, B. Deng, L. Guibas, Y. Zhou, D. Anguelov, GINA-3D: Learning to generate implicit neural assets in the wild, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), 2023, pp. 4913–4926, http://dx.doi.org/10.1109/CVPR52729.2023.00476.

[46] H. Xie, Z. Chen, F. Hong, Z. Liu, CityDreamer: Compositional generative model of unbounded 3D cities, 2023, arXiv preprint arXiv:2309.00610.