3D-SLNR: A Super Lightweight Neural Representation for Large-scale 3D Mapping

Chenhui Shi[1,2] Fulin Tang[1†] Ning An[3,4] Yihong Wu[1,2†]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]China Coal Research Institute [4]State Key Laboratory of Intelligent Coal Mining and Strata Control
shichenhui2022@ia.ac.cn,tangfulin2015@ia.ac.cn,ning.an@ccteg-bigdata.com,yhwu@nlpr.ia.ac.cn

Abstract

We propose 3D-SLNR, a new and ultra-lightweight neural representation with outstanding performance for large-scale 3D mapping. The representation defines a global signed distance function (SDF) in near-surface space based on a set of band-limited local SDFs anchored at support points sampled from point clouds. These SDFs are parameterized only by a tiny multi-layer perceptron (MLP) with no latent features, and the state of each SDF is modulated by three learnable geometric properties: position, rotation, and scaling, which make the representation adapt to complex geometries. Then, we develop a novel parallel algorithm tailored for this unordered representation to efficiently detect local SDFs where each sampled point is located, allowing for real-time updates of local SDF states during training. Additionally, a prune-and-expand strategy is introduced to enhance adaptability further. The synergy of our low-parameter model and its adaptive capabilities results in an extremely compact representation with excellent expressiveness. Extensive experiments demonstrate that our method achieves state-of-the-art reconstruction performance with less than 1/5 of the memory footprint compared with previous advanced methods.

1. Introduction

This paper aims to reconstruct large-scale scenes from point clouds, covering from indoors to outdoors, and even extending to city-scale environments. This is a key research topic in computer vision and graphics and remains greatly promising in applications such as self-driving, robotics, and mixed reality. However, the task presents significant challenges, as larger scene scales introduce more complex geometry and higher memory demands. Thus, an expressive yet highly compact map representation is desirable.

Previous approaches have extensively explored the optimal scene representations. Traditional approaches usually directly model scenes as discrete distance grids [28, 42] or predefined basis functions [13, 15], which then are converted into 3D fields through techniques like truncated signed distance function (TSDF) fusion [26] or finite element analysis [13, 15]. To enable scalability for large scenes, various sparse data structures [9, 12, 14, 28, 42] are widely employed. However, capturing detailed geometry with these methods necessitates a large allocation of data nodes, resulting in a substantial increase in memory usage. Building on the above traditional concepts, many learning-based methods [20, 37, 38, 48, 52] represent scenes as continuous and differentiable implicit fields using neural networks, often incorporating sparse feature grids [22, 25, 40] to facilitate scalability and expressiveness. Some methods [11, 46] reconstruct scenes using data-driven neural kernels, offering greater expressiveness than the basis functions used in traditional methods [13, 15]. These methods mitigate the need for extensive data node allocation but require numerous high-dimensional latent features to capture intricate geometric details, introducing additional memory overhead. The recent point-based learning methods [30, 35, 47] show great potential for lightweight representation due to their inherent adaptivity. However, they still heavily rely on latent features, neglecting the further exploitation of the adaptivity. In summary, prior methods often prioritize expressiveness at the expense of memory efficiency.

In this paper, we present 3D-SLNR, a new and super lightweight representation with exceptional expressiveness for large-scale 3D mapping. A visualization is provided in Fig. 1. Our method combines the principles from neural implicit representations and basis functions while fully exploiting the adaptive nature of the point-based representations. Specifically, we define a continuous and differentiable global SDF using a set of band-limited local SDFs anchored at support points sampled from point

† Corresponding author.

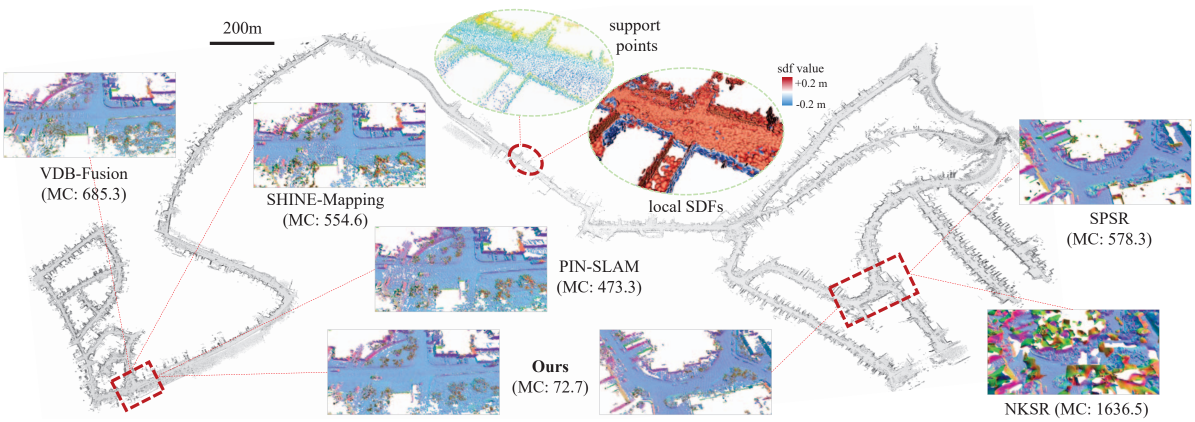

Figure 1. Our method defines a global SDF based on a set of band-limited local SDFs anchored at support points. It reconstructs more consistent and smoother meshes in a kilometer-scale scene in the KITTI-360 dataset with less than $1/5$ memory consumption (short in MC in megabytes) for map representation, compared to previous advanced methods.

clouds. Unlike the previous feature-dependent methods, we only parameterize these local SDFs with a small MLP, effectively serving as a learnable basis SDF, which significantly reduces the number of model parameters. To enhance expressiveness, we control the state of each local SDF by three learnable geometric properties—position, rotation, and scaling—allowing our representation to adapt flexibly to complex geometries. Furthermore, we introduce a novel parallel algorithm to efficiently detect the local SDFs where each sampled point is located, enabling real-time updates of the states of local SDFs during training. This capability is critical to making our unordered representation applicable to large-scale mapping. In addition, we propose a prune-and-expand strategy that removes redundant support points while allocating more points to under-reconstructed regions, further boosting adaptability. The above innovations make our representation extremely compact and strongly expressive. The main contributions of this paper are summarized as follows:

- A new and ultra-lightweight neural representation is proposed for large-scale 3D mapping that is feature-independent and highly expressive.
- A novel parallel local SDF detection algorithm tailored for the proposed representation is developed to enable real-time updates during training.
- A prune-and-expand strategy is introduced to enhance adaptability by dynamically controlling point allocation.
- Extensive experiments demonstrate that our method achieves state-of-the-art reconstruction performance with less than 1/5 of the memory footprint compared with previous advanced methods.

2. Related work

Traditional 3D reconstruction. Poisson reconstruction [13, 15] is one of the most popular methods. The central step is solving a Poisson equation to convert discrete points into a continuous indicator field. Specifically, it employs B-spline basis functions anchored at multi-resolution grid nodes [14] to fit the indicator function, followed by solving for the weights of each basis function. However, these basis functions are predefined, limiting their expressiveness. In complex scenes, they tend to fit well only in very localized regions, requiring a substantial increase in data nodes to improve expressiveness, which in turn leads to a large memory footprint. Another prominent approach is TSDF fusion-based reconstruction [26], which fuses per-frame TSDFs into a global representation. By leveraging sparse data structures such as octrees [9, 41], hash tables [12, 28, 29], and VDBs [42], reconstruction in large-scale scenes has been achieved. Nevertheless, significant numbers of data nodes still need to be assigned to accommodate intricate geometries. While our expressiveness comes from its adaptivity and the learnable basis SDF, avoiding excessive memory usage.

Neural implicit reconstruction. With the advent of seminal works such as Occupancy networks [23], DeepSDF [31], and NeRF [24], reconstruction methods based on neural implicit representations have rapidly emerged. Early approaches [1, 7, 27, 43, 50] only use single MLP architectures to represent scenes; however, the limited capacity of MLPs renders them inadequate for reconstructing large-scale scenes. To address this, a plethora of methods [17, 18, 44, 49] adopt sparse feature grids [25, 40] to enhance model capacity and improve geometric fidelity. Subsequently, methods like SHINE-Maping [52], along with [37], [48], [6], [38] and [32] demonstrate effective large-scale reconstruction utilizing ranging data. Furthermore, NKSR [11] reconstructs large-scale scenes using data-driven neural kernels [46] that are more expressive than predefined basis functions employed in Poisson methods [13, 15]. Despite these advancements, existing meth-

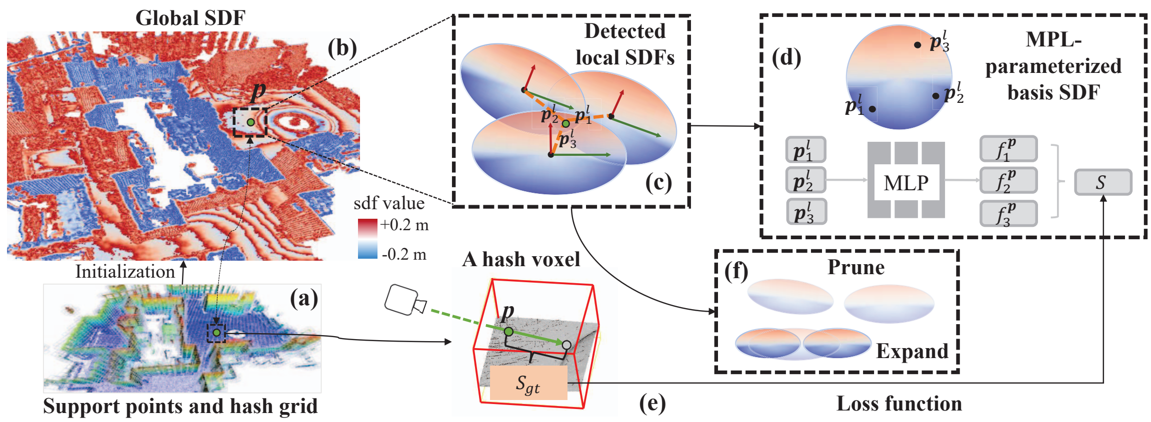

Figure 2. Overview of the proposed large-scale 3D mapping approach. (a) We construct a spatial hash grid near geometric surfaces and sample support points from the scene point cloud. (b) A global SDF constrained in the hash grid is initialized by setting up local SDFs anchored at these support points. (c) We employ our parallel algorithm to efficiently detect local SDFs where each sampled point is located. (d) The 3D coordinate of the sampled point is mapped from the world coordinate system to local coordinate systems defined by geometric parameters of the detected local SDFs. Through a tiny MLP, these local coordinates are converted into an SDF value. (e) We compute the SDF label for the point sampled in the hash grid and construct the loss function for training. (f) During training, a prune-and-expand strategy is performed at regular iterations.

ods for large-scale reconstruction heavily rely on numerous high-dimensional latent features, resulting in substantial memory overhead. In contrast, our approach eliminates the dependence on such latent features, significantly reducing memory consumption.

Point-based reconstruction. Surfel [33] is a type of classical point-based representation that consists of a set of oriented 2D disks attached to geometric surfaces. It has been widely adopted by 3D mapping methods [3, 5, 36, 45] due to its compactness and adaptability. However, this representation is discrete and unordered, limiting its ability to model continuous surfaces. Recent methods, such as Point-SLAM [35] and PIN-SLAM [30], represent scenes as continuous SDFs based on neural points, exhibiting promising results for 3D reconstruction. Yet, these methods still heavily depend on latent features and fail to leverage the natural adaptability of point-based representations fully. Approaches [10, 51] based on 3D Gaussian splatting (3DGS) [16] adopt efficient splatting and densification algorithms to ensure adaptive updates of point-based Gaussians; nevertheless, these algorithms are primarily designed for real-time image rendering, and their memory footprint escalates significantly during training. Our approach can be considered as a point-based continuous representation but does not rely on latent features. And, the introduced local SDF detection algorithm and prune-and-expand strategy are specifically developed for our representation, ensuring sufficient adaptability and efficiency.

3. Method

An overview of our approach is shown in Fig. 2. First, a spatial hash grid near geometric surfaces is constructed [37] and support points are sampled from the scene point cloud generated by accumulating posed point cloud frames. Then, we set up local SDFs anchored at these support points, creating an initial global SDF constrained in the near-surface space by the hash grid. Subsequently, our parallel algorithm is employed to efficiently detect the local SDFs at each sampled point in the hash grid. After that, we can obtain the SDF value of each sampled point through the geometric parameters of these detected local SDFs and a small MLP. Finally, we compute the SDF labels and construct the loss function for training. In the training process, a prune-and-expand strategy is performed at regular iterations. After training, we can extract meshes from the optimized global SDF by marching cubes algorithm [21]. In this section, we first describe our map representation in Sec. 3.1, and then introduce the local SDF detection algorithm in Sec. 3.2, followed by the prune-and-expand strategy in Sec. 3.3, and finally, the training process in Sec. 3.4.

3.1. Map representation

Our map representation, 3D-SLNR, defines a global SDF $\mathcal{F}^g$ that is composed of a series of local SDFs with the number of N:

$$\mathcal{F}^g = \{\boldsymbol{F}_i^l = (\mathbf{x}_i, \mathbf{r}_i, \mathbf{s}_i, F_\theta) \mid i = 1, ..., N\}, \qquad (1)$$

where each local SDF $\boldsymbol{F}_i^l$ is derived from a band-limited basis SDF F_θ with a radius of $m = 3$, parameterized by a tiny MLP with the parameter θ. And each local SDF is anchored at a support point with position $\mathbf{x}_i \in \mathbb{R}^3$ and shaped by a rotation vector $\mathbf{r}_i \in \mathbb{R}^3$ and a scaling vector $\mathbf{s}_i \in \mathbb{R}^3$. Therefore, the learnable parameters are $\{\mathbf{x}_i, \mathbf{r}_i, \mathbf{s}_i, \theta\}$, not including any latent features.

Given a sampled point $\boldsymbol{p}$ in the world coordinate system and its K detected local SDFs $\{\boldsymbol{F}_j^{\boldsymbol{p}} \mid j = 1, ..., K\}$, we can query the SDF value of $\boldsymbol{p}$ as presented in Fig. 2(d). Firstly, we transform $\boldsymbol{p}$ from the world coordinate system to the local coordinate system of the basis SDF through the geometric parameters of these detected local SDFs:

$$\boldsymbol{p}_j^l = \frac{\mathbf{R}_j^{-1}(\boldsymbol{p} - \mathbf{x}_j)}{exp(\mathbf{s}_j)}, \tag{2}$$

where $\mathbf{R}_j \in \mathbb{R}^{3\times 3}$ is a rotation matrix transferred by the rotation vector $\mathbf{r}_j$, and the exponentiation of $\mathbf{s}_j$ is to ensure that the scales of local SDFs are always greater than 0. Then, we can query the local SDF values from the MPL-parameterized basis SDF:

$$f_j^{\boldsymbol{p}} = F_\theta(\boldsymbol{p}_j^l). \tag{3}$$

Finally, the SDF value of $\boldsymbol{p}$ can be obtained by a weighted average:

$$S = \sum_{j=1}^{K} \frac{w_j}{\sum_{j=1}^{K} w_j} f_j^{\boldsymbol{p}}, \tag{4}$$

where w_j is a weight factor, which is calculated by:

$$w_j = exp(-||\boldsymbol{p}_j^l||_2^2). \tag{5}$$

The above formulation is continuous and differentiable for all parameters. As a result, we can update the distribution and shapes of these local SDFs during training, which endows our representation with high expressiveness and adaptivity.

According to the descriptions above, our approach can be viewed as a basis function-based neural implicit representation method. Compared to the predefined basis in traditional methods [13, 15], our basis SDF is data-driven and thus is more expressive. In comparison to the feature-derived kernels in learning-based methods [11, 46], ours is feature-independence and extremely compact. The proposed approach is also a point-based method but more adaptive than previous methods [30, 35, 47].

Initialization. We construct a spatial hash grid following [37], constraining the global SDF and point sampling to the near-surface space. The scene point cloud is voxel-downsampled with a voxel size of s_v to get the initial support points $\mathbf{x}^{init}$, which are also initial positions of local SDFs. The scaling vectors $\mathbf{s}^{init}$ of local SDFs are initialized as $[ln(s_v), ln(s_v), ln(s_v)]^T$. We perform principal component analysis (PCA) on neighbor points to estimate the normals $\mathbf{n}$ of support points and correct these normals toward the data acquisition sensor. Then, the initial rotation vectors $\mathbf{r}^{init}$ are derived by:

$$\mathbf{r}^{init} = \alpha \cdot \mathbf{r}_b,\ \alpha =< \mathbf{z}_{axis} \cdot \mathbf{n} >,\ \mathbf{r}_b = \mathbf{z}_{axis} \times \mathbf{n}, \tag{6}$$

where $\mathbf{z}_{axis} = [0, 0, 1]^T$. That makes $\mathbf{n}$ become the z-axis of the local coordinate system of each local SDF. The basis SDF F_θ^{init} is initialized to an implicit plane.

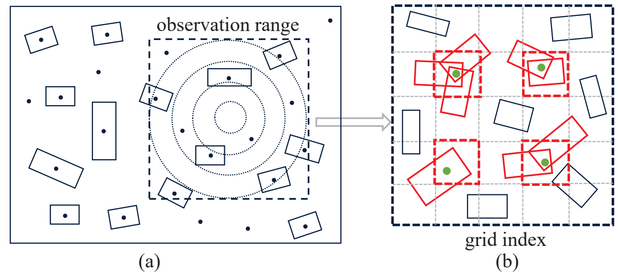

Figure 3. In an iteration, our parallel local SDF detection algorithm (a) first selects candidate local SDFs according to the current observation range, (b) then constructs a grid index suitable for parallel retrieval in the bounding box determined by these candidates, and finally detects local SDFs in the voxel where each sampled point is located in parallel.

3.2. Parallel local SDF detection for sample points

Fast detection of local SDFs where each sampled point is located is vital for the proposed unordered representation to enable large-scale 3D mapping. The common practice is to leverage special data structures like R-trees [2, 8], but their construction speed fails to meet the real-time demands of local SDF updates. Existing point-based methods, such as Point-NeRF [47] and Point-SLAM [35], utilize inefficient closest neighborhood searching, limiting them to small scenes. Although PIN-SLAM [30] enhances neighborhood querying through a hash index, it sacrifices the inherent adaptivity of point-based representations.

The recently rising 3DGS [16] has inspired us, which employs a highly efficient rasterizer for 3D Gaussians to achieve real-time rendering. Its core mechanism involves parallel detection of 2D Gaussians at the locations of regular image pixels in 2D screen space. In contrast, we focus on the parallel detection of 3D local SDFs at the locations of unordered sampled points in 3D space. Drawing on the insight of the similarity between these processes, we introduce a novel parallel algorithm specifically tailored for our representation. The algorithm is implemented using CUDA. We start with simplifying local SDFs as oriented bounding boxes determined by the geometric parameters $\{\mathbf{x}, \mathbf{r}, \mathbf{s}\}$ and the radius m, followed by selecting candidate local SDFs according to the current observation range of a sensor, such as a LiDAR or a depth camera, as shown in Fig. 3(a). Then, a regular grid is built covering these candidates. Subsequently, we instantiate each local SDF by signing keys for the candidates according to the IDs of intersected grid voxels. After that, these instances are sorted by their keys and stored in an array. We identify the first and last instances intersecting each grid voxel in the sorted array. At this stage, a grid index suitable for parallel retrieval is established, as illustrated in Fig. 3(b). Finally, for each sampled point in the observation range, we only need to detect the local SDFs in the voxel where the sampled point is located. As shown

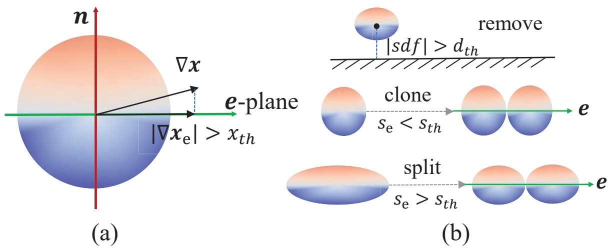

Figure 4. Prune-and-expand strategy.

in Fig. 3(b), local SDF detection is performed for four sampled points in parallel. An overview of our algorithm is presented in Suppl 1.

3.3. Prune-and-expand strategy

In this section, we introduce a prune-and-expand strategy to enhance the adaptivity of our representation further. First, we observe that the support points extracted from the scene point clouds can contain significant noise, such as that introduced by dynamic objects, and some may deviate from the geometric surface during training. This will compromise the accuracy of the reconstruction. Therefore, we remove the support points where the absolute values of their SDF values exceed a threshold d_{th} in the training process, as presented in Fig. 4(b). However, the subsequent problem is that removing the support points leads to a considerable loss of local SDFs, which hampers the completeness of the reconstruction. To address this, we aim to generate new local SDFs in the underfitting regions. We find that the local SDFs in these areas exhibit significant gradients $\nabla \mathbf{x}_e$ on planes (named $\boldsymbol{e}$) orthogonal to the normals of the support points, as illustrated in Fig. 4(a). If the norms $|\nabla \mathbf{x}_e|$ exceed a threshold x_{th} (set to 2×10^{-4}), new local SDFs will be created, as shown in Fig. 4(b). When the maximum scaling value s_e on the $\boldsymbol{e}$-plane is less than a threshold s_{th} (set to $0.8 s_v$), we clone a new local SDF along $\nabla \mathbf{x}_e$. And when s_e is larger than s_{th}, we split the local SDF into two smaller ones. The above process produces significantly different results from the densification strategy employed in 3DGS [16]. The densification of 3DGS leads to an extreme increase in Gaussians, whereas ours tends to remove more outlier local SDFs, resulting in reduced memory consumption (Tab. 4).

3.4. Training

As described in Sec. 3.1, a spatial hash grid has been constructed near the geometric surfaces. Then, we sample M points in the hash grid along rays [37], as shown in Fig. 2(e). Given a sampled point $\boldsymbol{p}$, an SDF value S can be queried from our representation. Subsequently, we obtain the SDF label S_{gt} according to the distance from the sampled point to the endpoint along the ray. Finally, the loss function is defined as follows:

$$L = \begin{cases} |S - S_{gt}| + \lambda(||\nabla_{\boldsymbol{p}} S||_2 - 1)^2, \ if \ |S_{gt}| < tr \\ |S - tr|, \ otherwise \end{cases}, \tag{7}$$

where tr is a truncation distance, λ is a weight factor set to 0.02. In fact, the loss function makes our representation become a truncated signed distance function (TSDF). During training, the prune-and-expand is performed every 200 iterations. For more efficient backward computing, we derive the gradients of Eq. 2 and implement them with CUDA (see Suppl. 2 for derivation details).

4. Experiments

Settings. We compare our approach with a wide range of 3D mapping methods, including VDB-Fusion [42], SPSR [13], NKSR [11], SHINE-Mapping [52] and PIN-SLAM [30], in large-scale indoor scenes and outdoor scenes. We adopt the commonly used metrics [39] named accuracy (Acc.) in centimeter (cm), completeness (Comp.) in cm, precision (Prec.) in percent (%), recall in %, and F-score in %, where Prec., recall and F-score are presented with a $10cm$ error threshold in indoor scenes and a $20cm$ error threshold in outdoor scenes. We also evaluate the memory consumption for map representation (MC) in megabytes (M). All experiments are conducted on a single RTX4090 GPU. More setting details are presented in Suppl. 3.

4.1. Evaluation in indoor scenes

We randomly select five scenes from a large-scale indoor RGB-D dataset, Matterport3D [4], as the experimental data. Tab. 1 presents the quantitative results in the dataset, where the values are averages of the results assessed in these scenes. Fig. 5 shows the results evaluated in a three-story loft (See Suppl. 4 for more results). The experimental results demonstrate that our approach achieves state-of-the-art 3D reconstruction performance in large-scale indoor scenes with less than 1/6 of memory consumption for map representation. Compared to NKSR and SHINE-Mapping, which depend on high-dimensional latent features, our approach eliminates this reliance, significantly reducing the memory footprint. Although VDB-Fusion and SPSR do not rely on latent features, they need to allocate a large number of data nodes to enhance the expressiveness, leading to a substantial increase in memory usage. In contrast, the expressiveness of our method comes from the adaptivity of our point-based representation and the data-driven basis SDF, avoiding excessive memory consumption.

4.2. Evaluation in outdoor scenes

We evaluate the 3D mapping performance of our approach in large-scale outdoor scenes on three LiDAR

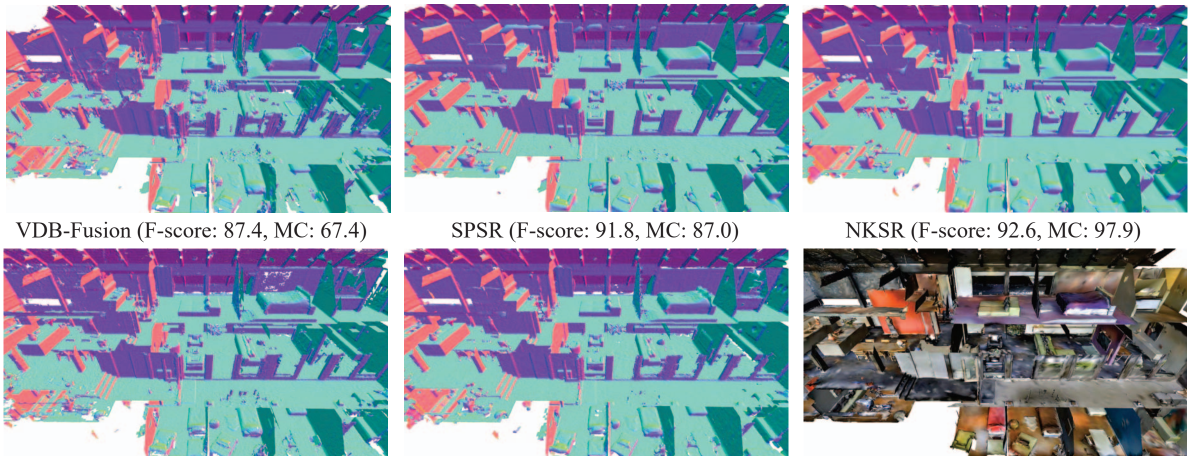

Figure 5. Experimental results evaluated in a three-story loft (about $11.8m \times 21.8m \times 11.7m$) in the Matterport3D dataset. The reconstructed meshes are colored by surface normals.

Method	Acc.↓	Comp.↓	Prec.↑	Recall↑	F-Score↑	MC↓
VDB-Fusion	4.6	5.4	90.3	92.5	91.3	112.5
SPSR	5.0	3.8	91.1	**96.8**	93.8	80.5
NKSR	4.7	3.8	92.1	96.5	94.2	128.9
SHINE-Mapping	4.9	3.7	90.1	95.8	92.8	167.4
Ours	**4.3**	**3.7**	**92.3**	96.5	**94.4**	**13.7**

Table 1. Quantitative results of different methods evaluated on the Matterport3D Dataset.

datasets, including the Newer College dataset [34], a self-collected street-level dataset (See Suppl. 3.1 for detailed description), and the KITTI-360 dataset [19]. The quantitative results listed in Tab. 2 demonstrate that our approach achieves state-of-the-art mapping performance (F-score) in the Newer College dataset while utilizing less than 1/5 of the memory footprint for map representation compared to the previous advanced methods. The best mapping performance (F-score) mainly benefits from the high accuracy, as Acc. and Prec. indicated. Fig. 6 shows the qualitative results evaluated in the scene of Math Institute in the Newer College dataset. VDB-Fusion, SPSR, and NKSR introduce artifacts associated with dynamic objects, as highlighted in the black dashed circles. SHINE-Mapping exhibits notable robustness against dynamic elements, resulting in cleaner meshes. Our method shares this advantage with SHINE-Mapping, while concurrently eliminating inconsistent meshes and achieving improved accuracy, as evidenced by the error maps. PIN-SLAM is sensitive to noise and yells uneven surfaces, which are also incomplete, as shown in the orange dashed box. In comparison, ours delivers a smoother result, primarily through the full utilization of the adaptability inherent in point-based representations.

Fig. 7 shows the results evaluated in a street scene in the street-level dataset. In this experiment, we adjust the memory footprint of other methods to be as close to that of ours as possible. VDB-Fusion produces many fragment meshes. The outputs of SPSR and NKSR tend to be overly smooth, resulting in a loss of crucial geometric details. This is particularly evident in NKSR, which shows an obvious decrease in F-score. PIN-SLAM has an F-sore close to ours, but its produced mesh is too noisy. While SHINE-Mapping can generate a visually plausible result, it significantly sacrifices accuracy, as indicated by the error maps. Our method achieves the best reconstruction performance due to its compact presentation and excellent expressiveness. See Suppl. 4.2 for more results.

The qualitative results evaluated on the KITTI-360 dataset are shown in Fig. 1. In the kilometer-level scene, our method continues to deliver the best mapping performance with less than $1/5$ memory footprint. The meshes generated by SPSR suffer from bulging, mainly due to its sensitivity to the accuracy of point normals, which tend to be poorly estimated from the noisy point cloud. NKSR yells extremely redundant meshes. VDB-Fusion introduces artifacts of dynamic objects. SHINE-Mapping and PIN-SLAM produce uneven road surfaces. In contrast, our method is more robust to normal noise and dynamic objects and produces more consistent and smoother results. See Suppl. 4.2 for more results.

Running time. As listed in Tab. 3, for 100 thousand sampled points, our parallel local SDF detection algorithm achieves millisecond-level running efficiency and our CUDA-improved backward computing is twice as fast as the native backward computing implemented in Pytorch, which facilitates the real-time updates of local SDFs.

4.3. Ablation study

Geometric parameters. As shown in Table 4, the update of geometric parameters significantly enhances map-

Method	Quad						Math Institute					
	Acc.↓	Comp.↓	Prec.↑	Recall↑	F-Score↑	MC↓	Acc.↓	Comp.↓	Prec.↑	Recall↑	F-Score↑	MC↓
VDB-Fusion	11.8	8.4	85.9	96.9	91.1	20.7	12.1	13.5	79.8	88.3	83.8	43.9
SPSR	11.2	9.7	87.8	96.1	91.8	39.6	11.6	31.8	80.6	85.5	83.0	112.3
NKSR	12.3	8.9	86.1	96.7	91.1	368.5	11.9	13.3	80.2	89.8	84.7	408.7
SHINE-Mapping	11.6	**7.8**	85.6	**97.5**	91.2	15.8	12.9	**9.9**	77.0	**92.5**	84.0	33.8
PIN-SLAM	13.3	8.8	82.7	94.6	88.3	18.8	16.6	12.7	68.1	84.4	75.4	42.9
Ours	**9.7**	9.7	**89.4**	96.3	**92.7**	**3.3**	**11.2**	20.3	**82.4**	88.5	**85.3**	**8.0**

Table 2. Quantitative results of different methods evaluated on the Newer College Dataset. The range of Quad is about $52.5m \times 71.3m \times 21.1m$ and the range of Math Institute is about $103.1m \times 109.0m \times 19.0m$.

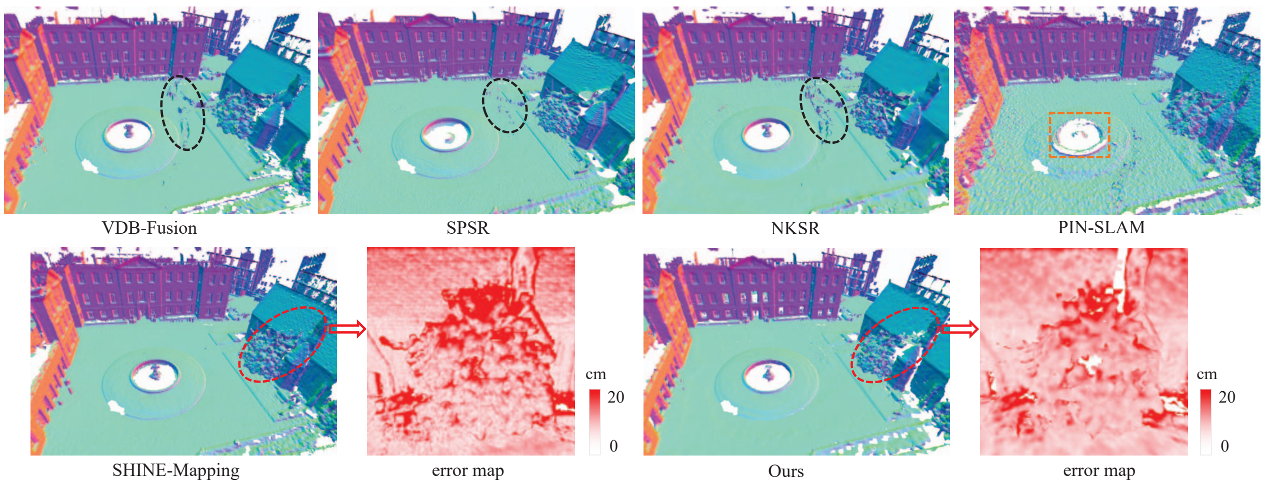

Figure 6. Qualitative results evaluated in the scene of Math Institute in the Newer College dataset. The reconstructed meshes are colored by surface normals. The error maps show the error distribution with the ground truth point cloud as a reference, where the redder areas mean larger errors up to $20cm$.

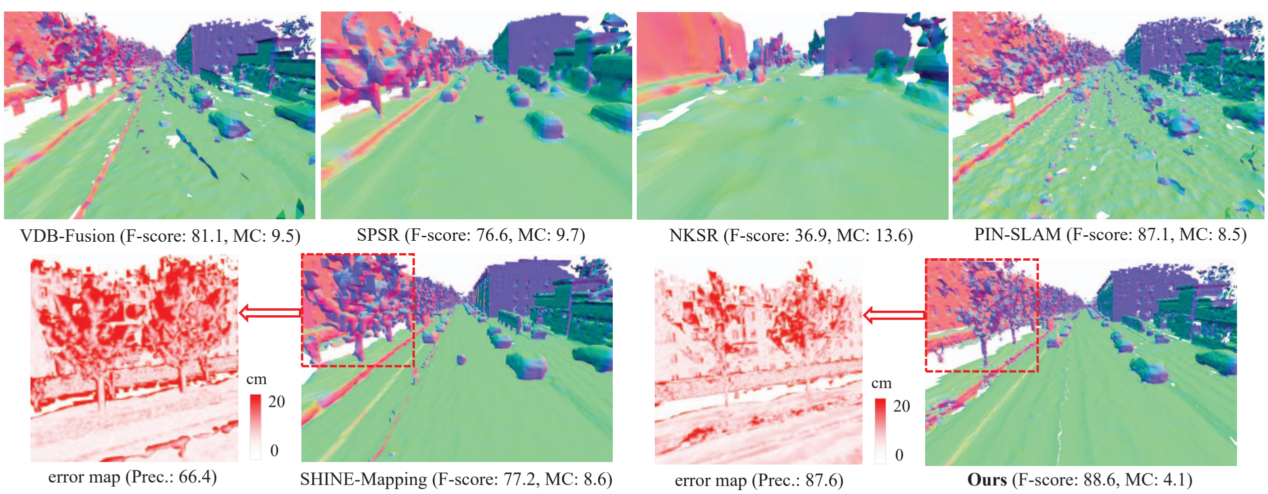

Figure 7. Experimental results evaluated in a street scene (about $235m$ long) in the street-level dataset. The reconstructed meshes are colored by surface normals. The error maps show the error distribution with the ground truth point cloud as a reference, where the redder areas mean larger errors up to $20cm$.

ping performance. With the sequential introduction of each geometric parameter, the F-score progressively improves, mainly due to a marked increase in precision. This incorporation fully leverages the adaptability of point-based representations, resulting in substantial performance gains in 3D mapping. These improvements are also attributed to our parallel local SDF detection algorithm, which enables the real-time updates of these geometric parameters.

Prune-and-expand. As shown in Table 4, our prune-and-expand strategy provides a boost to mapping perfor-

Process	local SDF detection	Backward (CUDA)	Total	Backward (Pytorch)
Time	8 ms	26 ms	75 ms	51 ms

Table 3. The running time of main modules in one iteration during training, which are evaluated in Math Institute in Newer College dataset for 100 thousand sampled points.

Method	Prec.↑	Recall↑	F-Score↑	MC↓
w/o geo	77.8	87.8	82.5	9.9
+rot	80.0	**90.0**	84.7	9.9
+rot+scale	81.5	88.8	85.0	9.9
+rot+scale+pos	82.3	88.4	85.2	9.9
Ours (+geo+pe)	**82.4**	88.5	**85.3**	**8.0**

Table 4. Quantitative ablation results for geometric parameters and prune-and-expand strategy evaluated in Math Institute. "geo" denotes all geometric parameters, "rot" denotes rotation parameters, "scale" denotes scaling parameters, "pos" denotes position parameters, and "pe" denotes prune-and-expand.

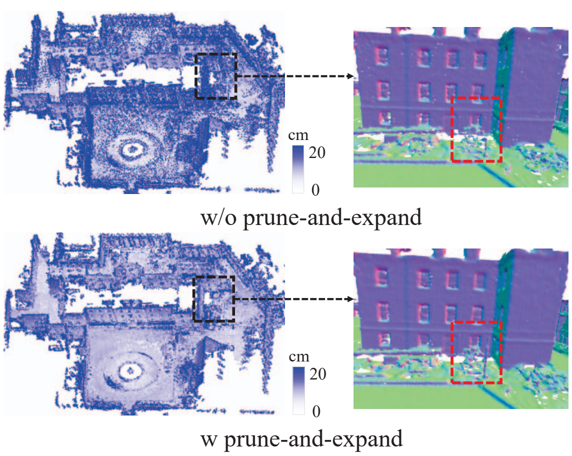

Figure 8. Qualitative ablation results for prune-and-expand evaluated in Math Institute. The left shows the error maps of support points, where the bluer points mean larger errors up to $20cm$. The right shows the close-ups of the reconstructed meshes.

mance in terms of both accuracy and completeness while concurrently reducing memory consumption. The reason is that the remove operation deletes most of the outlier support points, and clone and split operations complete the under-reconstructed regions, as presented in Fig. 8. This further enhances adaptability and expressiveness. The reduction in memory footprint results from the fact that more support points are removed than added, which significantly differs from the densification results of 3DGS [16]. According to the above results, our prune-and-expand strategy can be viewed as a form of parameter regularization.

Support point initialization. As illustrated in Fig. 9, we initialize support points by voxel-downsampling the scene point cloud with the voxel size $s_v = 1.0m$, $0.3m$, and $0.2m$, respectively. The memory footprint increases in order. Nevertheless, the F-score shows a trend of increasing and then decreasing. When the number of parameters is low, underfitting results will be produced. As shown in Fig. 9(a), the wall is too smooth and the ground has holes. When the number of parameters is high, overfitting results will be delivered. As shown in Fig. 9(c), the reconstructed wall is too noisy. When the number of parameters is moderate, the desired results can be generated, as shown in Fig. 9(b).

(a) s_v: 1.0m (b) s_v: 0.3m (c) s_v: 0.2m

Figure 9. Ablation results for support point initialization evaluated in Quad.

5. Conclusion and future work

In this paper, we have presented 3D-SLNR, a super lightweight neural representation with excellent performance for large-scale 3D mapping. The representation introduces a new map structure that leverages a set of band-limited local SDFs to define a continuous and differentiable global SDF near the geometric surfaces. Each local SDF is determined only by the MLP-parameterized basis SDF and three geometric parameters, eliminating the need for latent features and significantly reducing the number of parameters. Furthermore, we introduce a novel parallel local SDF detection algorithm tailored for our representation to ensure the real-time updates of the local SDFs. This allows our method to fully exploit its adaptivity, enhancing expressiveness. Additionally, we develop a prune-and-expand strategy to remove outlier support points and refine under-reconstructed regions, functioning as a form of parameter regularization and further improving overall mapping performance. Our method achieves state-of-the-art 3D mapping performance with an exceptionally low memory footprint, spanning various large-scale environments, from indoor and outdoor scenes to kilometer-level landscapes.

In future work, we will further optimize the running efficiency and integrate the representation into a simultaneous localization and mapping (SLAM) system for online city-scale reconstruction. Besides, exploring open-scene 3D generation based on our representation presents a potential research direction.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 62202468, Beijing Science and Technology Plan Project under Grant Z231100007123005, and Beijing Natural Science Foundation under Grant L241012.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 6290–6301, 2022. 2

[2] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*, pages 322–331, 1990. 4

[3] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In *Robotics: science and systems*, page 59, 2018. 3

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 5

[5] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE, 2019. 3

[6] Junyuan Deng, Qi Wu, Xieyuanli Chen, Songpengcheng Xia, Zhen Sun, Guoqing Liu, Wenxian Yu, and Ling Pei. Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8218–8227, 2023. 2

[7] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5511–5520, 2022. 2

[8] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, 1984. 4

[9] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34:189–206, 2013. 1, 2

[10] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[11] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 4369–4379, 2023. 1, 2, 4, 5

[12] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015. 1, 2

[13] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 1, 2, 4, 5

[14] Misha Kazhdan and Hugues Hoppe. An adaptive multi-grid solver for applications in computer graphics. In *Computer graphics forum*, pages 138–150. Wiley Online Library, 2019. 1, 2

[15] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 1, 2, 4

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3, 4, 5, 8

[17] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 30(3):1743–1755, 2022. 2

[18] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023. 2

[19] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 6

[20] Jianheng Liu and Haoyao Chen. Towards large-scale incremental dense mapping using robot-centric implicit neural representation. In *2024 IEEE ICRA*, pages 4045–4051. IEEE, 2024. 1

[21] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 3

[22] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 1

[23] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[24] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2

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

[26] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and track-

ing. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 1, 2

[27] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF ICCV*, pages 5589–5599, 2021. 2

[28] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ IROS*, pages 1366–1373. IEEE, 2017. 1, 2

[29] Yue Pan, Yves Kompis, Luca Bartolomei, Ruben Mascaro, Cyrill Stachniss, and Margarita Chli. Voxfield: Non-projective signed distance fields for online planning and 3d reconstruction. In *2022 IEEE/RSJ International Conference on IROS*, pages 5331–5338. IEEE, 2022. 2

[30] Yue Pan, Xingguang Zhong, Louis Wiesmann, Thorbjörn Posewsky, Jens Behley, and Cyrill Stachniss. Pin-slam: Lidar slam using a point-based implicit neural representation for achieving global map consistency. *arXiv preprint arXiv:2401.09101*, 2024. 1, 3, 4, 5

[31] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 2

[32] Minseong Park, Suhan Woo, and Euntai Kim. Decomposition of neural discrete representations for large-scale 3d mapping. In *ECCV*, pages 360–376. Springer, 2024. 2

[33] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000. 3

[34] 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 *2020 IEEE/RSJ International Conference on IROS*, pages 4353–4360. IEEE, 2020. 6

[35] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 1, 3, 4

[36] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 3

[37] Chenhui Shi, Fulin Tang, Yihong Wu, Xin Jin, and Gang Ma. Accurate implicit neural mapping with more compact representation in large-scale scenes using ranging data. *IEEE Robotics and Automation Letters*, 2023. 1, 2, 3, 4, 5

[38] Shuangfu Song, Junqiao Zhao, Kai Huang, Jiaye Lin, Chen Ye, and Tiantian Feng. N^{3}-mapping: Normal guided neural non-projective signed distance fields for large-scale 3d mapping. *IEEE Robotics and Automation Letters*, 2024. 1, 2

[39] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF conference on CVPR*, pages 15598–15607, 2021. 5

[40] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 11358–11367, 2021. 1, 2

[41] Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul HJ Kelly, and Stefan Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, 2018. 2

[42] Ignacio Vizzo, Tiziano Guadagnino, Jens Behley, and Cyrill Stachniss. Vdbfusion: Flexible and efficient tsdf integration of range sensor data. *Sensors*, 22(3):1296, 2022. 1, 2, 5

[43] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[44] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 2

[45] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, page 3. Rome, Italy, 2015. 3

[46] Francis Williams, Zan Gojcic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, and Or Litany. Neural fields as learnable kernels for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18500–18510, 2022. 1, 2, 4

[47] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 1, 4

[48] Dongyu Yan, Xiaoyang Lyu, Jieqi Shi, and Yi Lin. Efficient implicit neural reconstruction using lidar. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8407–8414. IEEE, 2023. 1, 2

[49] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 2

[50] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2

[51] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on CVPR*, pages 19447–19456, 2024. 3

[52] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations. In *2023 IEEE ICRA*, pages 8371–8377. IEEE, 2023. 1, 2, 5