2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Efficient Implicit Neural Reconstruction Using LiDAR

Dongyu Yan[1], Xiaoyang Lyu[2], Jieqi Shi[3], and Yi Lin[4]

***Abstract*— Modeling scene geometry using implicit neural representation has revealed its advantages in accuracy, flexibility, and low memory usage. Previous approaches have demonstrated impressive results using color or depth images but still have difficulty handling poor light conditions and large-scale scenes. Methods taking global point cloud as input require accurate registration and ground truth coordinate labels, which limits their application scenarios. In this paper, we propose a new method that uses sparse LiDAR point clouds and rough odometry to reconstruct fine-grained implicit occupancy field efficiently within a few minutes. We introduce a new loss function that supervises directly in 3D space without 2D rendering, avoiding information loss. We also manage to refine poses of input frames in an end-to-end manner, creating consistent geometry without global point cloud registration. As far as we know, our method is the first to reconstruct implicit scene representation from LiDAR-only input. Experiments on synthetic and real-world datasets, including indoor and outdoor scenes, prove that our method is effective, efficient, and accurate, obtaining comparable results with existing methods using dense input.**

I. INTRODUCTION

Research on 3D reconstruction and scene representation using implicit neural representation has received extensive attention. Representing volume density field [1], signed distance function (SDF) [2]–[4] or occupancy field [5]–[7] with a neural network, researchers manage to reconstruct high-quality 3D models with high resolution and low memory cost. Different input data have been used to optimize the implicit representation, including RGB images, depth images, and point clouds. Following NeRF [8], image-based implicit neural reconstruction methods [3], [6] use volume rendering to project 3D scenes to 2D and supervise with 2D photometric loss. However, RGB-only input can cause ambiguity due to occlusion, resulting in limited precision and noisy geometry. To overcome the limitations of RGB-only methods, some works [1], [4], [7] utilize depth information from multi-view stereo or depth cameras to assist supervision and avoid ambiguity. Nevertheless, such methods still add supervision to 2D depth images by volume rendering, resulting in weak supervision. Meanwhile, the use of color and depth cameras also require suitable light condition and has difficulty in large-scale and outdoor scenes. There are also methods that reconstruct directly using 3D global point cloud, demonstrating impressive results in accuracy [5], [9], [10]. However, these methods face other significant challenges before they can be widely applied. First, these reconstruction methods usually require ground truth spatial attributes as supervision. Also, the raw point cloud frames must be registered globally into world coordinate with accurate poses to avoid inconsistency, which introduces errors and complexity to the system. In the end, classic methods usually take hours to build an implicit representation, which lacks timeliness. The above problems prevent these methods from being widely used in various scenarios.

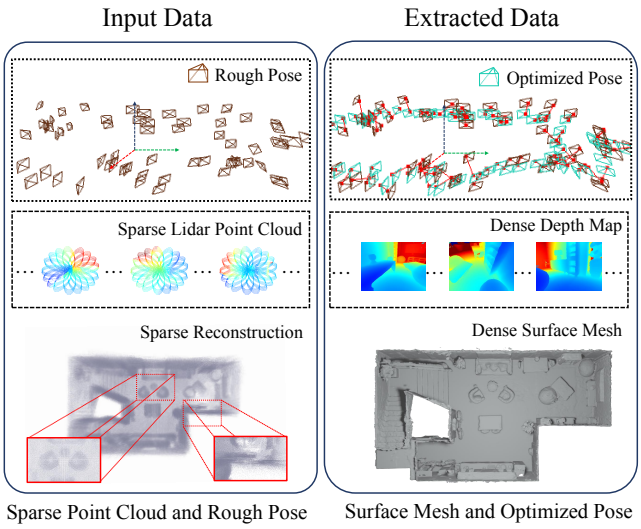

Fig. 1. Our method takes sparse LiDAR point clouds as input and outputs a dense occupancy field using implicit representation. We propose a new direct supervision method to handle the sparsity of the input data and refine the initial rough poses in a joint optimization module.

In this work, we explore the problem of optimizing an implicit occupancy field with only sparse LiDAR point clouds and set up a system that ensures accuracy, efficiency, robustness, and convenience at the same time. We point out the difficulty of the problem is that one LiDAR frame only contains around 5% depth data of a depth image. Such a difference indicates that depth rendering may not be able to provide enough geometry information for training and inspires us to seek a new way of supervision that better adapts to the sparsity of LiDAR point cloud. Leveraging the property of laser traveling, we propose to supervise directly in 3D space and manage to get rid of volume rendering and ground truth labels. With our object-thickness assumption, the new direct supervision achieves unbiased and occlusion-aware. Besides, we manage to free our method from cumbersome global registration by treating each frame

[1]Dongyu Yan is with School of Mechanical Engineering and Automation, Harbin Institute of Technology (Shenzhen). 21s053072@stu.hit.edu.cn
[2]Xiaoyang, Lyu is with Department of Electrical and Electronic Engineering, The University of Hong Kong. xylyu@eee.hku.hk
[3]Jieqi Shi is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. jshias@connect.ust.hk
[4]Yi Lin is with Dji Co. ylinax@connect.ust.hk

individually, which avoids blending them into a global point cloud. Such modification enables us to use rough poses provided by odometry as initial and refine them along with the optimization of the scene representation [11], [12]. Moreover, we refer to Instant Neural Graphics Primitives (Instant-NGP) [13] and use a multi-resolution hash encoder to accelerate the reconstruction process and provide fine-grained local and global geometry efficiently within a few minutes.

In summary, we present a method that takes sparse LiDAR point clouds and rough odometry as input and optimize an implicit occupancy field for reconstruction. Our method works efficiently and achieves good reconstruction results much faster than rendering-based methods. Also, due to the adaptability of LiDAR measurement, our method is suitable for a wide range of scenarios, including large-scale outdoor scenes. Our main contributions are as follows:

- We propose a new loss function that enables dense implicit reconstruction from sparse LiDAR inputs;
- We add pose refinement along with scene optimization to further decrease initial pose error and create consistent geometry;
- Our implemented reconstruction method is efficient and only consumes a few minutes per scene.

II. RELATED WORK

a) Classical Method: The field of 3D reconstruction has been well explored using classical methods. Researchers reconstruct 3D models from different kinds of input data, including RGB-only [14], [15], RGB-D [16]–[18] and LiDAR point cloud [19]. RGB-only methods exploit photometric consistency between images and optimize 3D information under geometry constraints. Methods utilizing depth data from depth camera or LiDAR directly obtain 3D information and build a global point cloud map. However, noisy depth measurement may create inconsistent geometry. To overcome this, fusion-based methods [16], [18] propose to maintain a global volumetric map containing geometry information such as signed distance or occupancy, which can be used to fuse and refine noisy depth data. By this means, 3D reconstruction tasks can be processed globally by optimization in a continuous way.

b) Learning-based Method: Classical methods may lose efficacy when faced with poor light conditions or intense depth noise. Learning-based methods utilize photometric and geometric priors to perform robust reconstruction to reduce these effects. Some works [20]–[22] utilize learned features to replace hand-craft features for better correlation detection. Others abandon the traditional feature matching process and build a cost volume to mimic the multi-view stereo matching using 3D CNN [23]–[25]. Moreover, geometry priors are leveraged to solve problems of noisy depth values [26] and sparse depth input [27]. However, there are still drawbacks to these methods. The use of CNN forces them to represent scenes discretely, creating limited precision and huge memory cost. Also, the domain gap can be a constraint to their generalization ability.

c) Implicit Neural Representation of Geometry: Representing geometry with implicit neural networks has recently gained much attention for its high spatial resolution and low memory cost. In contrast to traditional 3D representations, it models scenes with a continuous function containing attributes such as volume density, signed distance, or occupancy in its 3D coordinate. With the help of geometry labels of spatial coordinates, implicit representation can be directly learned with global inputs [2], [5], [9]. However, it is hard to obtain ground-truth labels in real-world scenes. Therefore, researchers have put forward several methods to train the network without the strict requirements for labels. For example, SAL [28] and SA-ConvONet [29] leverages signed agnostic learning and Controlling Neural Level Sets [30] takes another method to control the decision boundary directly. Furthermore, other methods [31], [32] propose to decompose shape into local parts and use voxel-grids to store scene information. Optimization of the implicit function can also be reached by local supervision using color or depth images. NeRF [8] presents a volume rendering method to supervise implicit representation by 2D photometric loss. However, the use of the volume density has caused rough surfaces. To this end, methods introducing volume rendering into SDF [3], [4], [33], [34] or occupancy [6], [7] representations have been proposed and achieves better surface reconstruction. Other methods fuse depth information into the frame work to further constraint optimization [35]–[38].

However, as we have mentioned before, such methods either assume known spatial information or require a suitable environment. In this paper, we hope to relax restrictions on practical applications and explore an implicit reconstruction method more suitable for various scenarios and usages, leveraging the use of LiDAR.

III. METHOD

A. System Overview

In this work, we present a new implicit neural reconstruction method using sparse LiDAR point clouds. Our method takes point cloud frames as input and optimizes an occupancy field. We show the pipeline of our method in Fig. 2. Given input point cloud sequence and initial poses from odometry, we first select key frames according to the change of viewpoint for removing redundant frames and filter out poor depth measurements. We then sample points from the key frames and use them to train our implicit occupancy field. We use a multi-layer perceptron (MLP) with a multi-resolution hash encoder as our implicit model(Section III-B). Furthermore,

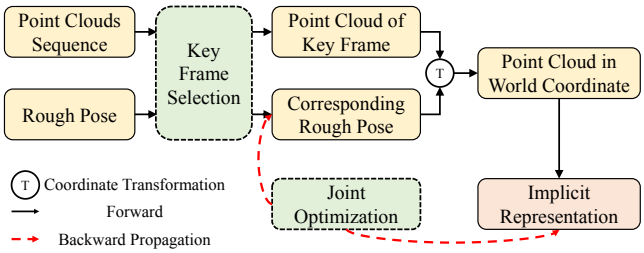

Fig. 2. Pipeline of our method.

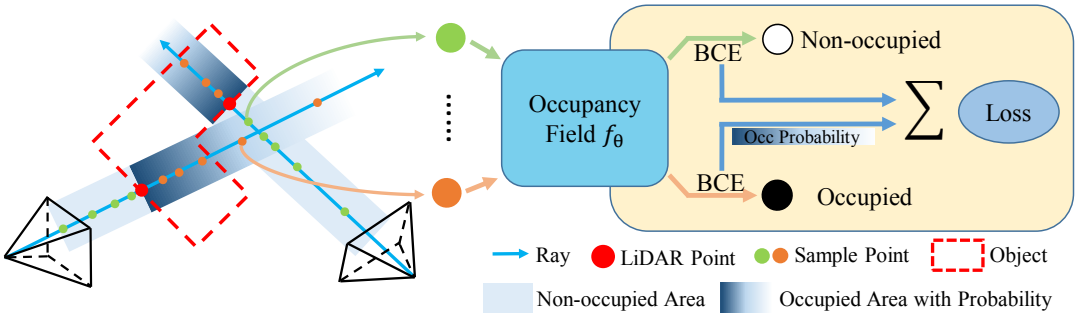

Fig. 3. We use LiDAR depth measurement to separate a ray into two sides. We supervise the occupancy of sample points directly to occupied or non-occupied using a BCE loss. To make our direct loss occlusion aware, we add probability to occluded points using our thickness assumption. The final weighted BCE loss achieves unbiased and occlusion-awareness.

We propose a direct supervision method to fully utilize depth information from sparse point cloud inputs (Section III-C). Finally, we jointly optimize our implicit network and sensor poses.(Section III-D). In the end, the dense surface mesh is extracted from our optimized occupancy field using Marching Cubes [39].

B. Implicit Representation with Hash Encoder

Our network means to fit an implicit function $f_\theta(\mathbf{p}) : R^3 \rightarrow [0,1]$ to the scene's occupancy field $\mathcal{O}(\mathbf{p}) : R^3 \rightarrow \{0,1\}$, which maps a spatial coordinate to its occupancy property. Since we only have sparse point clouds as input, the supervision of the implicit function is relatively sparse too. Under such condition, a low-frequency-aware structure is needed to interpolate areas without supervision. On the other hand, high-frequency information also needs attention for a fine-grained reconstruction. Therefore, we follow Instant-NGP and utilize a multi-resolution hash table to encode our coordinates to balance both low and high-frequency parts of the scene.

The hash table is built upon multi-resolution voxel-grids of the scene. We use a spatial hash function to link learnable features with voxel coordinates. For a spatial point in the 3D space, its feature embedding can be obtained by trilinear interpolation. We then concatenate the feature vectors from all resolutions to form the frequency-aware multi-resolution latent vector. The use of the hash table can decrease memory cost to a great extent. However, hash collision may occur in high-resolution layers, which may cause artifacts in under-supervised areas. To avoid this, we extract an additional feature vector from the input coordinate and concatenate them as our final feature.

Since the hash table has stored most scene features, we use a shallow MLP with low capacity to decode the features and get the occupancy decision boundary with a sigmoid function. Our complete network structure is shown in Fig. 4.

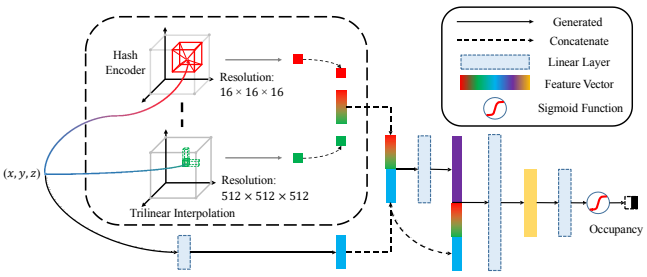

Fig. 4. Implicit network architecture of our method.

C. Direct Supervision

In order to fully utilize the view information of each key frame, we treat points in the point cloud as rays. Specifically, given a point cloud frame containing n points as $\{\mathbf{p}_i \mid i = 1, \cdots, n\}$, where each point is a vector of (x_i, y_i, z_i) coordinate, the points on the ith ray can be represent as $\widetilde{\mathbf{p}}_i(z) = \mathbf{o}_i + z\mathbf{d}_i$, where $\mathbf{o}_i$ is the origin of the sensor and $\mathbf{d}_i$ is the direction of the ray.

According to the mechanism of the LiDAR measurement, the laser can travel on the ray before hitting $\mathbf{p}_i$, which implies that the occupancy value along the ray before $\mathbf{p}_i$ should be 0, and the value behind $\mathbf{p}_i$ should be 1. In this way, a simple binary cross-entropy (BCE) loss can be directly applied to the occupancy function.

$$\mathcal{L}_d(\widetilde{\mathbf{p}}_i(z)) = \begin{cases} -\log(1 - f_\theta(\widetilde{\mathbf{p}}_i(z))), & \text{for } z < z_i \\ -\log(f_\theta(\widetilde{\mathbf{p}}_i(z))), & \text{for } z \geq z_i \end{cases}. \quad (1)$$

However, due to occlusion, we can not obtain object thickness and thus can not determine the range of the area be supervised as occupied. Simply treating the occluded space as occupied leads to ambiguity when the same region is observed by other views as non-occupied.

To solve the occlusion problem and give supervision to points sampled behind objects, we make a simple object-thickness assumption. We propose a definition called generalized thickness. The generalized thickness b refers to the distance of a single continuous occupied area that a ray $(\mathbf{o}, \mathbf{d})$ travels through:

$$b = min(z_s - z_i \mid \mathcal{O}(\mathbf{o} + z_s\mathbf{d}) = 0.5), \quad (2)$$

where $\mathcal{O} = 0.5$ means the level-set of the surface and z_s means the depth of the very next surface the ray intersects with. Based on this definition, we treat the generalized thickness of objects in the environment as a random variable B. We assume that it obeys a logarithmic normal distribution: $\ln(B) \sim \mathcal{N}(\mu, \sigma^2)$. Relying on this prior information of the scene's geometry, the probability of a point $\widetilde{\mathbf{p}}_i(z)$ sampled on the ray with $z \in (z_i, z_s)$ being occupied can be derived

$$P_{occ}(z) = P(B > z - z_a) = 1 - F_B(z - z_a), \quad (3)$$

where F_B means the cumulative distribution function of the variable B. Once the occupancy probability of the occluded

sample point is obtained, supervision can be applied to the points behind the object. At the same time, a probability of 1 is given to the non-occupied supervision. By weighting the simple BCE loss function with the probability of observation, an occlusion-aware direct loss on a point $\widetilde{\mathbf{p}}_i(z)$ can be modified as follows:

$$\mathcal{L}_d(\widetilde{\mathbf{p}}_i(z)) = \begin{cases} -\log(1 - f_\theta(\widetilde{\mathbf{p}}_i(z))), & \text{for } z < z_i \\ -P_{occ}(z)\log(f_\theta(\widetilde{\mathbf{p}}_i(z))), & \text{for } z \geq z_i \end{cases} \quad (4)$$

With the occlusion-aware direct loss, the ambiguity can be solved, and a clean implicit neural representation can be optimized.

D. Joint Optimization

As we already have the surface position on each ray, to focus more on the near surface areas, we use a normal distribution to sample m points on each ray: $\{\widetilde{\mathbf{p}}_i(z_j) \mid j = 1, \cdots, m\}, z_j \sim \mathcal{N}(z_i, \sigma_s^2)$, where σ_s is a hyper parameter related to scene size. We additionally sample another k points in a stratified way. During optimization, we first sample a batch of N_b points from N_f point clouds and form $N_f \times N_b$ rays. Then we sample $m + k$ points on each ray and use these total $N_f \times N_b \times (m + k)$ points for supervision.

Besides the loss function $\mathcal{L}_d(\widetilde{\mathbf{p}}_i(z))$ mentioned in Section III-C, we add another surface regularization loss $\mathcal{L}_n(\mathbf{p}_i)$ on surface points to encourages a smooth surface. The normal loss we use has the following format:

$$\mathcal{L}_n(\mathbf{p}_i) = |1 - \mathbf{n}(\mathbf{p}_i) \cdot \mathbf{n}(\mathbf{p}_i + \epsilon)|, \quad (5)$$

where ϵ is a small neighbor range and $\mathbf{n}(\mathbf{p}_i)$ represents the normal of the implicit function at $\mathbf{p}_i$, denoted as

$$\mathbf{n}(\mathbf{p}_i) = \frac{\nabla_{\mathbf{p}_i} f_\theta(\mathbf{p}_i)}{\|\nabla_{\mathbf{p}_i} f_\theta(\mathbf{p}_i)\|_2}. \quad (6)$$

Our final loss function is defined as

$$\mathcal{L} = \frac{1}{N_f N_b} \sum_{i=0}^{N_f N_b} \left(\frac{1}{m+k} \sum_{j=0}^{m+k} \lambda_d \mathcal{L}_d(\widetilde{\mathbf{p}}_i(z_j)) + \lambda_n \mathcal{L}_n(\mathbf{p}_i) \right). \quad (7)$$

Since rays are transformed to world coordinate by poses of each frame, we can trace the gradient using backpropagation. Then we refine the initial poses while optimizing the implicit representation simultaneously by minimizing $\mathcal{L}$. This way, errors of the rough poses generated by the odometry system can be further reduced, creating a more consistent 3D geometry.

IV. EXPERIMENTS

A. Implementation Details

We implement our hash encoder following the setting in Instant-NGP. During optimization, we use ADAM [40] optimizer with a learning rate of 1×10^{-3} for both the implicit function and poses. The loss weight is set to $\lambda_d = 1.0$ and $\lambda_n = 0.4$. Every iteration, we sample points for supervision using $N_b = 100$, $m = 32$ and $k = 8$. The value of σ_s is set to 0.3 which is suitable for most scenarios. We run 300 iterations per scene with around 5 minutes on a single NVIDIA RTX2080Ti GPU. After 150 iterations, we decrease the learning rate of pose refinement to 1×10^{-4} to focus more on reconstruction. After optimization, we discretize our implicit function into voxel-grids of resolution 512 and extract mesh using Marching Cubes. To cull the mesh in the unseen area, we filter out vertices 0.1 meters away from our registered point cloud.

B. Experimental Settings

a) Datasets: We perform quantitative evaluation on synthetic datasets, including 9-Synthetic-Scenes [1] [36] and Replica [41]. We also test our method on real-world datasets of ScanNet [42] and our self-collected outdoor dataset.

b) Baselines: (1)*COLMAP:* We run COLMAP [14] to generate poses and register a point cloud from depth images. We then use Screen Poisson surface reconstruction [43] to generate mesh. (2)*BundleFusion:* We feed dense color and depth maps directly to BundleFusion [18] for poses and geometric reconstruction. (3)*ConvONet:* We use poses generated by BundleFusion and accumulate dense point clouds from the input depth map. We then run Convolutional Occupancy Network [10] on the global point cloud using a pre-trained network provided by the authors. (4)*NeRF with depth loss:* We add additional depth loss to NeRF [8], where rendered depth is compared with input depth map using L2 distance. The surface is extracted on the volume density field. (5)*Neural RGB-D:* We use dense color and depth maps to perform Neural RGB-D [36] scene reconstruction.

C. Results on Synthetic Datasets

We perform quantitative evaluation on 9-Synthetic-Scenes dataset. We evaluate the reconstructed mesh against ground truth using Chamfer ℓ_1 distance and F-score [44] metrics with the threshold of 0.05 meter. We use the settings mentioned in Section IV-B following Neural RGB-D. To simulate a sparse LiDAR input, we randomly sample 15000 points per frame from the ground truth depth value and treat them as input data. We then use BundleFusion to generate initial poses. As shown in Table I, our method receives comparable results with only sparse input (about 5% of a depth map of 640×480 resolution).

We also evaluate the efficiency of our method, and the results are shown in Table II. On average, our method requires fewer training frames and sparser rays per frame, resulting in less training time than Neural RGB-D while performing comparably. With our key frame selection scheme, hash

[1]The other 9 scenes tested in Neural RGB-D except ICL data.

TABLE I
QUANTITATIVE RESULTS ON 9-SYNTHETIC-SCENES DATASET. FOR FAIRNESS, METRICS WITH * REMOVE "THE KITCHEN" SCENE AT EVALUATION FOR CORRUPTION DATA.

Method	Sensor	C ℓ_1*	F-score*	C ℓ_1	F-score
COLMAP	RGB-D	0.036	0.835	0.060	0.743
BundleFusion	RGB-D	0.046	0.809	0.067	0.788
ConvONet	RGB-D	0.050	0.680	0.073	0.658
NeRF-D	RGB-D	0.045	0.782	0.070	0.762
NeuralRGB-D	RGB-D	**0.023**	**0.941**	0.048	0.917
Ours	**LiDAR**	0.030	0.921	**0.030**	**0.920**

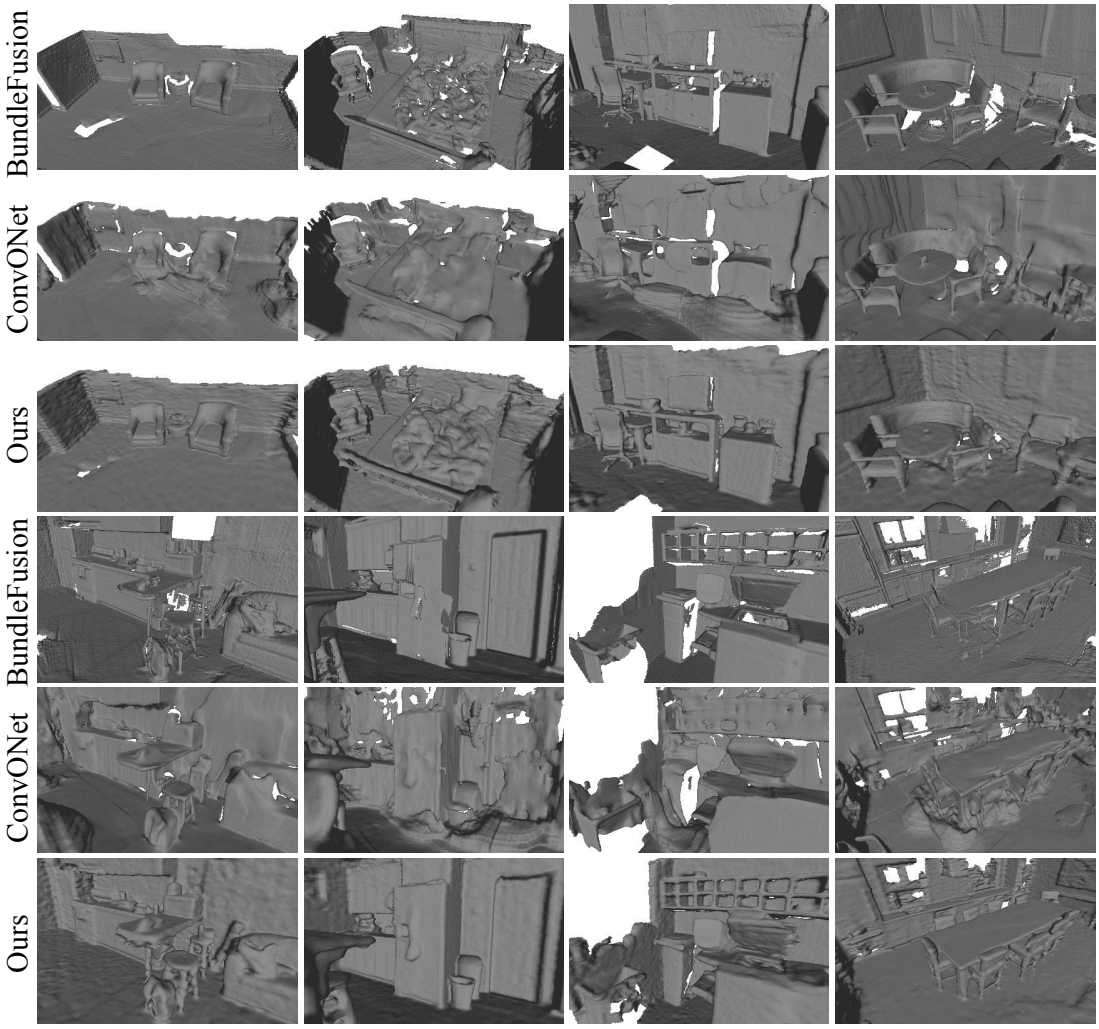

Fig. 5. Qualitative results compared with BundleFusion and ConvONet on SacnNet dataset.

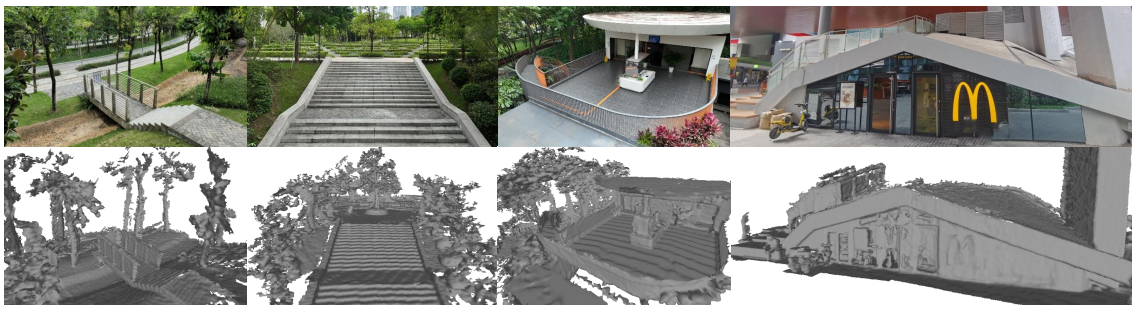

Fig. 6. Qualitative results in large-scale outdoor environment. We show that our method is adaptable to various scenarios.

encoder, and direct loss, we have saved training time to a great extent.

TABLE II

TIME CONSUMPTION RESULTS.

Method	Neural RGB-D	Ours
Average Frames Used	1219	118
Rays Per Frame	307200	15000
Average Time	540 min	**5 min**
Time Per Frame	26.6 s	**2.5 s**

We test the pose refinement ability of our method on eight synthetic indoor scenes from the Replica dataset. Poses are initialized by adding noise to the ground truth. We add uniform distribution of $U(-0.05, 0.05)$ radius to rotation and $U(-0.1, 0.1)$ meter to translation. The generated mesh result can be seen in Fig. 8. It shows that our direct supervision can optimize poses with noise and achieves consistent results.

D. Results on Real-World Datasets

We employ ScanNet to operate experiments on real-world data. Similar to Section IV-C, we randomly sample 15000 points on the depth map as the pseudo LiDAR input. In Fig. 5, we compare our method to the original BundleFusion reconstructions and ConvONet. It can be seen that our model can generate a continuous surface and fill the holes that

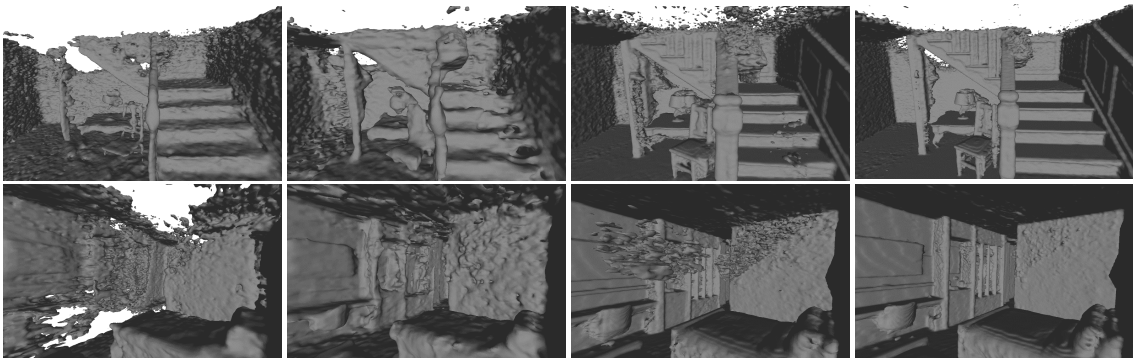

Fig. 7. Quantitative results of ablation studies.

TABLE III

QUANTITATIVE RESULTS OF ABLATION STUDIES.

Method	depth render loss		w/o pose refinement		w/o thickness prior		ours full model	
Metrics	**C-ℓ_1**	**F-score**	**C-ℓ_1**	**F-score**	**C-ℓ_1**	**F-score**	**C-ℓ_1**	**F-score**
9-Synth	0.098	0.514	0.044	0.553	0.036	0.867	**0.030**	**0.920**
Replica	0.155	0.341	0.049	0.693	0.030	0.879	**0.025**	**0.934**

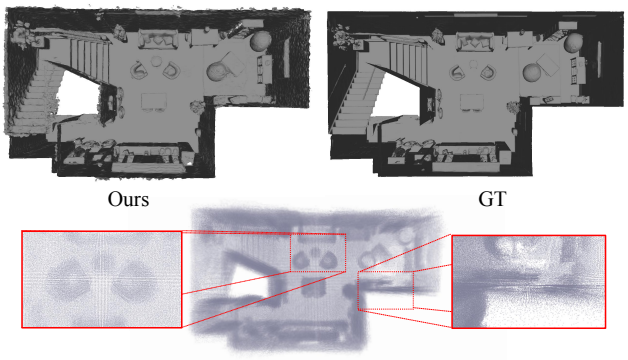

Fig. 8. Qualitative results on Replica dataset. We evaluate the pose refinement ability of our method by giving noisy initial poses. The bottom image shows the accumulated global point cloud, which is blurry due to inaccurate initial poses.

appeared in the BundleFusion models. We also solve the misalignment problems in ConvONet by our pose alignment module and generate a fine-grained surface with sparse input.

Apart from the public datasets, we manage to test our framework using a commercial LiDAR in real world. We build up an outdoor LiDAR scan dataset using Livox AVIA, which generates 24000 points per frame at 10 fps. We run Fast-LIO [45] to obtain initial poses. As shown in Fig. 6, our method works well in large-scale outdoor scenes with great movement. In this way, our framework further proves its practicability and broad application prospect.

E. Ablation Studies

We analyze the effect of each module in our framework, including direct loss, pose refinement, and thickness prior, by replacing or removing them from our framework one at a time. The complete results of the ablation study can be found in Fig. 7 and Table III

a) Direct Loss vs. Depth Render Loss: As we have mentioned, direct loss enables us to supervise on sparse point clouds in much less training time and is one of the critical contributions of our framework. Here we replace the direct loss with the prevalent L2 depth rendering loss and compare the results in two synthetic datasets.

b) Effect of Pose Refinement: Sensor poses obtained from the odometry can be drifting and noisy. Directly using these poses will cause inconsistency in geometry. We test our method by evaluation after removing the pose refinement module.

c) Effect of the Thickness Prior: We analyze the effect of the thickness prior by removing the occlusion-aware term from the loss function and using the simple BCE loss (Equation 1) to guide the training.

Table III shows that all three modules play an essential role in building our reconstruction framework. The direct loss function ensures that we can supervise the reconstruction well with sparse input data, which relaxes the strict requirements on input data and helps reduce the training time. The pose refinement module refines the initial poses jointly and ensures the continuity and accuracy of the final result. The thickness prior solves the ambiguity of occupancy in the occluded space and removes the artifacts. We believe that the experiments once again prove the rationality and innovation of our framework.

V. CONCLUSION

In this paper, we put forward an implicit 3D reconstruction method using only sparse point cloud frames from a LiDAR. We use implicit representation with the hash encoder and a newly proposed direct loss to deal with the sparsity of input data and suggest a thickness assumption to handle the occlusion problem. The experiments show that with around 5% of the input data, we can achieve comparable reconstruction results with the methods using dense input. Moreover, our framework requires only a few minutes for training, significantly reducing time consumption. The limitation of our method is that the hand-tuned thickness prior lacks generalization. In the future work, we will explore to solve this by adapting learnable priors.

REFERENCES

[1] E. Sucar, S. Liu, J. Ortiz, and 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.

[2] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[3] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021.

[4] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "isdf: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[5] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[6] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5589–5599.

[7] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," *arXiv preprint arXiv:2112.12130*, 2021.

[8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[9] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5939–5948.

[10] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 523–540.

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

[12] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5741–5751.

[13] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[14] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[15] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[16] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.

[17] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense rgb-d mapping," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5724–5731.

[18] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[19] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and 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)*. IEEE, 2017, pp. 1366–1373.

[20] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[21] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[22] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3247–3257.

[23] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 767–783.

[24] Z. Murez, T. v. As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3d scene reconstruction from posed images," in *European Conference on Computer Vision*. Springer, 2020, pp. 414–431.

[25] J. Sun, Y. Xie, L. Chen, X. Zhou, and H. Bao, "Neuralrecon: Real-time coherent 3d reconstruction from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 598–15 607.

[26] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 175–185.

[27] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 103–119.

[28] M. Atzmon and Y. Lipman, "Sal: Sign agnostic learning of shapes from raw data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2565–2574.

[29] J. Tang, J. Lei, D. Xu, F. Ma, K. Jia, and L. Zhang, "Sa-convonet: Sign-agnostic optimization of convolutional occupancy networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6504–6513.

[30] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman, "Controlling neural level sets," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, "Deep local shapes: Learning local sdf priors for detailed 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2020, pp. 608–625.

[32] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser *et al.*, "Local implicit grid representations for 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[33] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.

[34] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[35] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, "Dense depth priors for neural radiance fields from sparse input views," *arXiv preprint arXiv:2112.03288*, 2021.

[36] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," *arXiv preprint arXiv:2104.04532*, 2021.

[37] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," *arXiv preprint arXiv:2107.02791*, 2021.

[38] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," *arXiv preprint arXiv:2111.14643*, 2021.

[39] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[42] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[43] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[44] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.
[45] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.