2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Towards Large-Scale Incremental Dense Mapping using Robot-centric Implicit Neural Representation

Jianheng Liu and Haoyao Chen

***Abstract*— Large-scale dense mapping is vital in robotics, digital twins, and virtual reality. Recently, implicit neural mapping has shown remarkable reconstruction quality. However, incremental large-scale mapping with implicit neural representations remains problematic due to low efficiency, limited video memory, and the catastrophic forgetting phenomenon. To counter these challenges, we introduce the Robot-centric Implicit Mapping (RIM) technique for large-scale incremental dense mapping. This method employs a hybrid representation, encoding shapes with implicit features via a multi-resolution voxel map and decoding signed distance fields through a shallow MLP. We advocate for a robot-centric local map to boost model training efficiency and curb the catastrophic forgetting issue. A decoupled scalable global map is further developed to archive learned features for reuse and maintain constant video memory consumption. Validation experiments demonstrate our method's exceptional quality, efficiency, and adaptability across diverse scales and scenes over advanced dense mapping methods using range sensors. Our system's code will be accessible at https://github.com/HITSZ-NRSL/RIM.git.**

I. Introduction

Large-scale dense mapping is integral to various autonomous robotics tasks, encompassing autonomous driving, surveying, and inspection. Storing geometric information in grid maps is common in robotics and 3D vision. Traditional mapping methods efficiently construct the map using diverse data structures like octree [1], VDB [2], and hash map [3]. Recent works employ range sensors to realize efficient, high-quality reconstructions, like Voxblox [4] and VDBFusion [5]. Although these methods guarantee good geometric accuracy, they struggle with high-granularity representation and scene speculation. Conversely, the recent developments in implicit neural representations [6], [7] leverage multi-layer perceptrons (MLPs) to model 3D scene structures and advance in high-fidelity representation and scene speculation. More specific geometry attributes, like occupancy [8] and SDF [9], are introduced to delineate surfaces avoiding geometry ambiguities [10]. iSDF [11] train an MLP to correlate 3D coordinates with an approximate signed distance using a series of posed depth images. These pure MLP-based methods depict small scenes with tiny memory, but MLP's limited capacity constraints a detailed and broader scene depiction.

This work was supported in part by the National Natural Science Foundation of China (Grant No.U21A20119 and No.U1713206) and in part by the Shenzhen Science and Innovation Committee (Grant No.JCYJ20200109113412326 and No.JCYJ20210324120400003). (Corresponding author: Haoyao Chen.)

J.H. Liu and H.Y. Chen* are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, P.R. China. liujianheng@stu.hit.edu.cn, hychen5@hit.edu.cn.

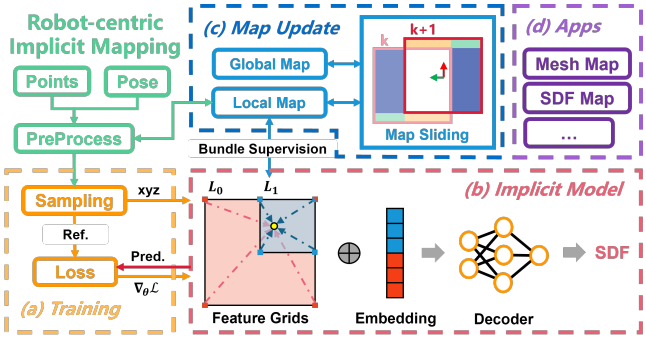


Fig. 1. Pipeline of the robot-centric implicit mapping. RIM generates signed distance fields in real-time based on provided posed points.

Explicit geometric structures [12]–[14] are integrated to embed learnable features in voxels to better sculpture scenes at the expense of memory consumption. However, extending the implicit neural representations to large-scale scenes is challenging due to the limited video memory. SHINE-Mapping [15] harnesses the memory-efficient octree to reduce video memory consumption but still struggles to reconstruct vast scenes using mainstream devices with limited video memory.

Nonetheless, most implicit neural representations are trained in a batch manner, which is unsuitable for instant tasks. Incremental mapping supports streaming data processing, offering instantaneous feedback [16] with greater flexibility than batch processing. However, incremental implicit neural mapping, as a continual learning problem, encounters the catastrophic forgetting phenomenon [17], wherein neural networks forget previously acquired knowledge during new learning phases. The idea of keyframes is introduced to hold a bundle adjustment for a consistent mapping [11], [18], [19]. SHINE-Mapping [15] employs a regularization-based approach to mitigate this challenge by confining the update direction of learnable features. However, designing well-generalized keyframe selection strategies and selecting regularization parameters is tricky.

Reconstructing a large-scale dense map necessitates precise and comprehensive geometric information. Although some implicit neural representations [7] can reconstruct high-fidelity synthetic views from RGB images alone, they need much effort to present precise geometry. Range sensors effectively capture dense environmental geometrics and are extensively employed for pose estimation [20], detailed mapping [21] and motion planning [22], [23]. The incorporated depth information [24] also contributes to faster convergence and more refined geometric details in implicit neural repre-

sentations. Our method follows the dense mapping paradigm using range sensors.

To address the challenges posed by the MLP's limited capacity, escalating video memory consumption, catastrophic forgetting, and dynamic objects, we propose robot-centric implicit mapping (RIM) for large-scale incremental dense mapping. The overall pipeline is depicted in Fig. 1. We embed learnable features in voxels and use a shallow MLP to infer signed distance field (SDF) for high-granularity representation, similar to voxel-based implicit representations [13]. We propose robot-centric mapping that trains the implicit model only on the local map with constant video memory consumption. It integrates numerous smaller local map training tasks into the whole global map training. To enable unbounded mapping, RIM crafts a global map using a hash map and sub-map set as a shelf to store and retrieve learned features from and for the local map. To mitigate the impact of catastrophic forgetting and dynamic objects, RIM leverages historical points within the local map to realize a multi-view bundle supervision training and outlier removal.

This paper's main contributions are threefold:

1) We propose a novel training framework for large-scale implicit neural representation using robot-centric mapping with constant video memory consumption.
2) We exploit the traits of robot-centric mapping to conduct bundle supervision and outlier removal, achieving high-quality reconstruction and mitigating the catastrophic forgetting and dynamic object influence.
3) We construct a flexible global map structure that allows dynamic unbounded map expansion without needing a preset map boundary.

Validation experiments are performed to verify the superior quality, efficiency, and versatility of our method across varied scales and scenarios.

Algorithm 1: Overall Frame Integration Process

Input: Current position $\mathbf{x}_k$, Input point cloud $\mathcal{P}_k$
Notation: Inside point cloud $\mathcal{P}_i$, Outside point cloud $\mathcal{P}_o$, Historical point cloud $\mathcal{P}_h$, Supervision set $\mathcal{S}$

Algorithm
  `Slide`($\mathbf{x}_k$); // Sec.II-B.1
  $\mathcal{P}_h, \mathcal{P}_o$ = `PreProcess`($\mathcal{P}_k$); // Sec.II-B.2
  foreach *iteration* **do**
    $\mathcal{S}$ = `Sample`($\mathcal{P}_h, \mathcal{P}_o$); // Sec.II-B.3
    `Train`($\mathcal{S}$); // Sec.II-A.2
  end
End Algorithm
Function `Slide`($\mathbf{x}, \mathcal{P}$)
  `SaveOutsideBlockToGlobal`($\mathbf{x}$);
  `UpdateLocalMapCenter`($\mathbf{x}$);
  `PadOutsideBlockFromGlobal`($\mathbf{x}$);
End Function
Function `PreProcess`($\mathcal{P}$)
  $\mathcal{P}_i, \mathcal{P}_o$ = `SeperateInOutsidePoint`($\mathcal{P}$);
  `OutlierRemoval`($\mathcal{P}_h$);
  $\mathcal{P}_h$ = `UpdateHistoricalPoint`($\mathcal{P}_i$);
  return $\mathcal{P}_h, \mathcal{P}_o$;
End Function
Function `Sample`($\mathcal{P}_\mathrm{h}, \mathcal{P}_\mathrm{o}$)
  $\mathcal{S}_h$ = `SampleHistoricalPoint`($\mathcal{P}_h$);
  $\mathcal{S}_o$ = `SampleOutsidePoint`($\mathcal{P}_o$);
  return $\{\mathcal{S}_h, \mathcal{S}_o\}$;
End Function

II. Methodology

Our work aims to reconstruct a high-granularity large-scale dense map from a continuous stream of poses and depth data. To accomplish this, we present our implicit neural representation for signed distance fields, which is trained in Euclidean space without the need for space compaction [11]. Subsequently, we describe our proposed robot-centric mapping method for high-quality implicit model training. The comprehensive frame integration process is shown in Alg.1, with each step elaborated in subsequent sections.

The notations are defined as follows: G, L denote the global and local coordinate systems; $\mathbf{x}, \mathbf{p}$ represent the robot's and point's positions, respectively.

A. Implicit Neural Representation

1) Implicit Model: We adopt a hybrid representation that encodes shape with learnable features ξ and decodes SDF values $\hat{d}$ using a shallow multi-layer perceptron, as illustrated in Fig. 1(b). The local map is composed of uniform feature voxels, where each voxel contains a learnable feature of dimensionality D. We adopt a multi-resolution feature matrix with L layers to integrate information at multiple granularities [13]. In this approach, the voxel size of the l-th layer map is determined by $2^{l-1}s$, where s is the setting leaf voxel size. Given a point $\mathbf{p}_i^G \in \mathbb{R}^3$ that falls into the local map, we obtain its different-levels embedding feature $\xi_l(\mathbf{p}_i^G) \in \mathbb{R}^D$ by trilinear interpolation with its eight neighboring features. Different levels of implicit feature ξ_l are concatenated into a new feature $\xi \in \mathbb{R}^{L \times D}$ integrated with multi-granularity information. An MLP-based decoder f_θ passes forward the concatenated feature and returns the predicted SDF value $\hat{d}_i$:

$$\hat{d}_i = f_\theta \left(\bigoplus_{l=1}^{L} \xi_l \left(\mathbf{p}_i^G \right) \right), \tag{1}$$

where $\bigoplus$ denotes the concatenation operation. We use $\hat{d}_i = f_\theta(\mathbf{p}_i^G)$ for brevity in the following sections.

2) Training: As illustrated in Fig. 2, given any input point $\mathbf{p}$ and its ray direction $\mathbf{r}$, we sample points along its ray: $\bar{\mathbf{p}} = \mathbf{p} - d\mathbf{r}$, and form the supervision set $\mathcal{S} = \{\mathbf{p}, d\}$. Each sample point's reference value d defines the signed ray distance, which may differ from the true SDF value. Inspired by [15], we use the sigmoid function to map SDF to range $(0, 1)$: $S(d) = 1/(1 + e^{d/\sigma})$, where σ is a hyperparameter to characterize the reconstruction smoothness and the sensor noise, and the 3σ distance can be considered as a soft

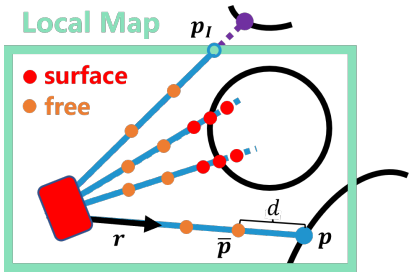


Fig. 2. The schematic diagram of spatial sampling. The high-quality input within the perceptual range carves the surface in detail, while the outside input distinguishes the free space. Moreover, the local map dynamically maintains historical points for bundle supervision.

truncated distance. The binary cross entropy is employed as the loss function as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i}^{N} o_i \log(\hat{o}_i) + (1 - o_i)\log(1 - \hat{o}_i), \tag{2}$$

where $o_i = S(d_i)$ is the reference value, $\hat{o}_i = S(f_\theta(\mathbf{p}_i))$ is the implicit model's predicted value, and N is the number of supervision points. The combination of the sigmoid function and the binary cross entropy adaptively gives higher weights to points near surfaces. The implicit model parameters are updated according to gradients $\nabla_\theta \mathcal{L}$ of the loss function (Alg.1-Ln.6), as shown in Fig. 1(a).

Random values are commonly used for parameter initialization to enhance learned features' generalization [15]. In this paper, we initialize all voxel features with zeros and the decoder with random values, expecting features to have similar implicit representations of the same shapes so that the decoder can have a consistent input to ease its burden.

B. Robot-centric Mapping

1) Map Structure: In this paper, the implicit model is only trained on a robot-centric local map whose center aligns with the robot's position regardless of rotation. The local map's size is set according to the sensor's perceptual range and the robots' task space size. The local map slides the dense feature matrix to the new origin according to the robot's position without destroying or allocating memory (Alg.1-Ln.11), as shown in Fig. 1(c) 's map sliding. A global map stores the learned features for reuse and is decoupled from the local map. It consists of sub-maps and a shared decoder with the local map. A sub-map is a feature matrix the same size as the local map. As shown in Fig. 3, every grid represents a sub-map, and the global map is covered with sub-maps without overlapping.

The local map is a cut of the global map, as shown in Fig. 3, where each voxel in the local map has a unique mapping voxel in the global map:

$$\mathbf{v}_i^G = \mathbf{v}_i^A - \mathbf{v}_L^A + \mathbf{v}_L^G, \tag{3}$$

where $\mathbf{v}$ represents the voxel index and A denotes local map matrix coordinate system. With the local map moving to an unexpanded area, a new sub-map is created and stored in the global map using a hash table. When the origin of

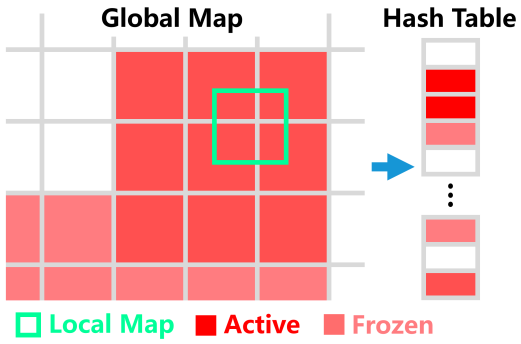


Fig. 3. The schematic diagram of the implicit map structure and the sub-maps management. The local map slides in the global map. Each grid represents a sub-map, and those without color are non-allocated. A hash table arranges the allocated sub-maps. Red grids indicate active sub-maps stored in video memory, and pink grids indicate frozen sub-maps stored in system memory.

the local map moves, blocks of voxels go out of the local map; and these blocks' features are cached to save into the corresponding sub-maps in the global map using a separate thread (Alg.1-Ln.10). The newly expanded block is padded with fetched features from the global map (Alg.1-Ln.12).

The robot-centric map structure decouples the training and storage maps using local and global maps. It allows us to reconstruct a large-scale scene with constant video memory. As shown in Fig. 3, we use a freeze-activate mechanism to transfer sub-maps between video and system memory. We keep the local map's nearby sub-maps activated for fast feature access. When a sub-map is far from the current pose, we freeze it by moving it to the system memory. When the robot revisits a historically frozen sub-map, we reactivate it by moving it back to video memory.

2) Bundle Supervision for Incremental Mapping: Incremental neural mapping as a continual learning method suffers from catastrophic forgetting problems [17]. Voxel-based implicit representations mitigate this problem by decoupling part of implicit features using the explicit geometric representation; however, due to the property of shared features between voxels, the historical features still might be degenerated by adjacent new inputs [15].

To address this problem, we separate the input point cloud into the inside point cloud $\mathcal{P}_i$ and outside point cloud $\mathcal{P}_o$ according to the local map's scope (Alg.1-Ln.15). The inside point cloud $\mathcal{P}_i$ is accumulated into historical point cloud $\mathcal{P}_h$ to conduct bundle supervision in training (Sec. II-A.2). We discard historical points that slide out of the local map and randomly drop exceeding points to keep a constant point number for fixed memory usage (Alg.1-Ln.17).

The proposed bundle supervision is more like point cloud registration than keyframe-based methods, introduces multi-views to constrain features, and maintains point rays instead of view attitudes. It does not need delicate keyframe selection strategies or keyframe maintenance in keyframe-based methods [11] or training efficiency sacrifice in regularization-based methods [15]. However, the decoder's parameters also update continuously during the training, which can cause inconsistency between learned features over time. Therefore,

TABLE I

QUANTITATIVE RECONSTRUCTION RESULTS OF 8 SCENES ON THE REPLICA DATASET. OUR APPROACH YIELDS THE BEST RECONSTRUCTION ACCURACY (C-L1) AND COMPLETENESS (F-SCORE) OVER OTHER METHODS.

Metrics	Methods	Office-0	Office-1	Office-2	Office-3	Office-4	Room-0	Room-1	Room-2
Acc.[cm]↓	Voxblox	1.08	0.85	1.03	1.06	0.93	0.91	0.80	1.04
	VDBFusion	**0.56**	**0.53**	**0.56**	**0.58**	**0.58**	**0.59**	**0.54**	**0.56**
	iSDF	2.02	2.07	2.07	2.01	1.65	1.86	1.31	2.20
	SHINE	2.23	1.67	2.31	2.15	1.48	2.01	2.45	1.99
	Ours	1.12	0.93	0.99	1.06	0.83	1.01	0.85	0.94
Comp.[cm]↓	Voxblox	10.35	9.70	5.19	3.16	3.43	3.09	2.92	4.22
	VDBFusion	9.75	9.07	4.71	2.97	3.18	2.96	2.81	3.79
	iSDF	12.90	11.89	14.16	3.74	5.55	3.09	2.72	9.02
	SHINE	**8.57**	7.68	4.23	2.89	3.01	2.58	2.25	3.44
	Ours	8.61	**7.49**	**3.78**	**2.45**	**2.65**	**2.26**	**2.14**	**3.09**
C-L1[cm]↓	Voxblox	5.71	5.27	3.11	2.11	2.18	2.00	1.86	2.63
	VDBFusion	5.15	4.80	2.64	1.77	1.88	1.77	1.68	2.17
	iSDF	7.46	6.98	8.12	2.87	3.60	2.48	2.01	5.61
	SHINE	5.40	4.68	3.27	2.52	2.24	2.30	2.35	2.72
	Ours	**4.86**	**4.21**	**2.39**	**1.75**	**1.74**	**1.63**	**1.46**	**2.02**
F-Score[%]↑	Voxblox	91.15	89.35	92.14	94.26	94.10	95.43	95.58	93.81
	VDBFusion	91.80	90.33	93.21	94.64	94.62	95.35	95.40	94.68
	iSDF	88.67	88.02	86.42	92.65	91.36	95.03	96.16	89.18
	SHINE	88.70	89.86	91.59	93.54	93.54	94.47	90.85	92.90
	Ours	**92.49**	**91.81**	**94.44**	**95.71**	**95.67**	**96.65**	**96.90**	**95.76**

we fix the decoder parameters after learning a certain number of frames.

3) Saptial Sampling and Outlier Removal: Supervision points are sampled from the surface and the free (non-occupied) space along the ray to train a consistent implicit map. As shown in Fig. 2, we sample historical inside points and current input outside points for each training iteration (Alg.1-Ln.5). For inside point rays in $\mathcal{P}_i$, we use depth-guided sampling: N_s surface points' sampling follows the normal distribution $\mathcal{N}(d, \sigma^2)$ and N_f free points are stratified sampled between surfaces and the sensor (Alg.1-Ln.21). For outside point rays in $\mathcal{P}_o$, we calculate their intersections $\mathbf{p}_I$ (Fig. 2) with the local map and stratified sample points between intersections and the sensor (Alg.1-Ln.22).

The free and outside point supervision helps RIM to mitigate the influence of dynamic objects and outliers by further applying outlier removal to the historical point cloud (Alg.1-Ln.16). The proposed outlier removal periodically infers the SDF values of historical points and drop points whose SDF values are ϵ away from 0, where the zero level set of the SDF defines the fitting surface. In the following training, the previously learned outlier features will be forgotten with the free and outside point supervision, as shown in Fig. 8.

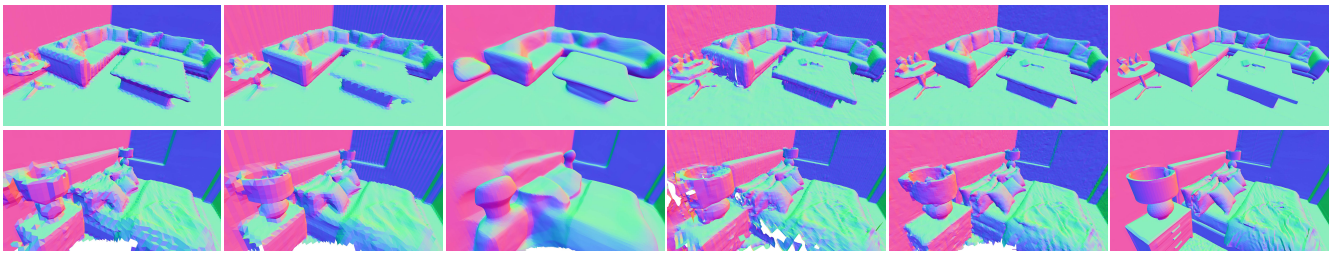
III. EXPERIMENTS

This section conducts qualitative and quantitative experiments on public datasets to demonstrate the novelty and effectiveness of the proposed methods. We compare our method with state-of-the-art dense mapping methods: Voxblox [4] and VDBFusion [5] based on traditional mapping methods, and iSDF [11] and SHINE-Mapping [15] based on implicit neural representations; and all methods reconstruct scenes incrementally with the same voxel size. We recover the implicit model to triangular mesh using marching cubes [25]. Since the proposed method cannot map surfaces outside the local map, we use a cropped ground truth map for a fair comparison. The widely used reconstruction metrics are used for quantitative evaluation [6], [15]: they are mapping accuracy (Acc., cm), completeness (Comp., cm), chamfer-L1 distance (C-L1, cm), and F-score (<10cm, %).

The RIM experimental parameters, namely L, D, ϵ, N, N_s, and N_f, are set to 3, 8, 5cm, 2048, 3, and 3, respectively. The MLP used in our experiments has only a single hidden layer with dimension 32 with ReLU activations in the intermediate layer. We form our implicit model using Libtorch and train it using the Adam optimizer. All experiments are conducted on a platform equipped with an Intel i5-12600KF CPU with 32GB system memory and an NVIDIA RTX 3070 Ti GPU with 8GB video memory.

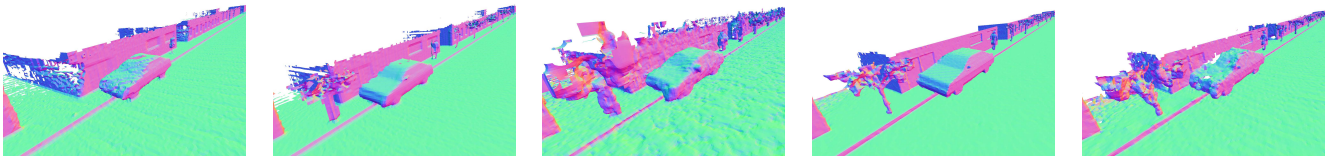
A. Map Quality

1) Replica Dataset: The Replica dataset [26] provides indoor simulation data generated by a camera mounted on a mobile manipulator. This experiment uses 8 of its scenarios for validation of indoor scene reconstruction. For small scenes, the leaf voxel size s is set to 0.05m, and σ is set to 0.02. The quantitative reconstruction results are shown in Tab. I. Regarding accuracy, VDBFusion performs best among all methods, with the lowest accuracy values for all scenes. Regarding completeness, SHINE performs best among all methods except for our proposed approach, with the lowest completeness values for most scenes. The pro-



(a) Voxblox (b) VDB-Fusion (c) iSDF (d) SHINE-Mapping (e) Ours (f) Ground Truth

Fig. 4. Reconstructed mesh on the Replica dataset's office-2 and room-1, and colors indicate the direction of the surface normal. Our method can capture richer, more complete, and precise geometric details in small objects like tables, pillows, and quilts.



(a) Voxblox (b) VDB-Fusion (c) SHINE-Mapping (d) Ours (e) Ours (w/o bundle)

Fig. 5. Reconstructed mesh on the MaiCity dataset sequence 01, and colors indicate the surface normal direction. Our method outputs smoother and more complete mapping results, such as the trees, car, walker, walls, and the ground.

TABLE II

QUANTITATIVE RECONSTRUCTION RESULTS ON THE MAICITY DATASET. AND ABLATION STUDIES FOR THE PROPOSED METHOD'S VALIDATION

	Methods	Acc.↓	Comp.↓	C-L1↓	F-Score↑
	Voxblox	2.554	2.875	2.715	97.261
	VDBFusion	**2.091**	1.382	1.736	98.596
	SHINE	3.189	1.854	2.521	95.472
	Ours	2.167	**0.958**	**1.562**	**98.923**
a)	random init.	2.982	1.118	2.050	96.082
b)	w/o bundle	2.485	1.247	1.866	98.229
c)	w/o free	2.217	0.963	1.590	98.723
d)	w/o outside	2.213	0.988	1.600	98.835
e)	Ours(online)	2.347	1.133	1.740	98.747

posed approach strikes a good balance between accuracy and completeness, yielding the best overall performance in terms of C-L1 and F-Score. The qualitative results from Fig. 4 also indicate that explicit geometric representations show a sharper outline but lack completeness, and the learning-based implicit representations can speculate an unseen surface and output continuous mapping results at the expanse of details, especially for the pure neural-network-based method, iSDF.

2) MaiCity Dataset: We evaluate the urban scenario on sequence 01 of the MaiCity dataset [27] that provides 100m noise-free 64-line lidar data and a ground truth model of a synthetic city. For large scenes, the leaf voxel size s is set to 0.1m, and σ is set to 0.05. iSDF is not included as it is not scalable for large scenes. The quantitative and qualitative comparisons of different methods on the MaiCity dataset are shown in Tab. II and Fig. 5, respectively. It should be aware that SHINE-Mapping results come from its official incremental mapping with a regularization strategy, which differs from the results in its original paper [15] using offline batch processing. Our proposed method can produce smoother and more complete reconstruction results while retaining more fine-grained information. In particular, our method can generate flat surfaces on geometric structures, such as walls and floors, and also capture rich details on small objects, such as tree trunks and car rear-view mirrors. Voxblox and VDBFusion show decent accuracy but sacrifice completeness, like trees. However, SHINE-Mapping using a regularization-based method still faces forgetting issues in large-scale incremental mapping, resulting in inconsistent reconstruction results.

3) Ablation Study: To verify the effectiveness of the contributions, we conduct ablation studies and show the results in Tab. II. As described in Sec. II-A.2, **(a)** the random initialization of implicit features badly hinders the decoder from learning consistent signed distance fields. **(b)** Without bundle supervision (Sec. II-B.2), both the reconstruction accuracy and completeness drop significantly, as shown in Fig. 5**(e)**. The free **(c)** and outside **(d)** point supervisions (Sec. II-B.3) slightly improve the mapping quality but help in mitigation of dynamic object influence (Sec. III-D).

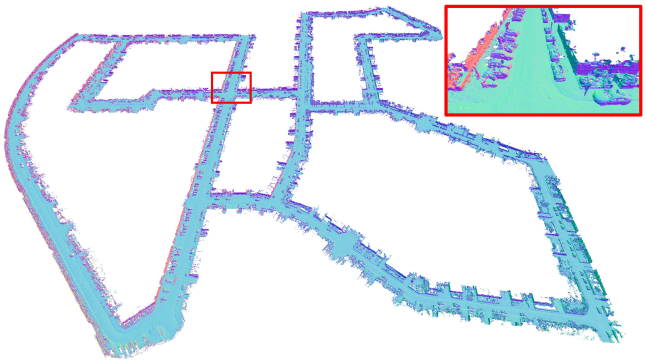


Fig. 6. Reconstruction results on sequence 00 of the KITTI dataset, and the red box shows a local zoom-in view. Our proposed method can reconstruct high-quality, large-scale scenes using constant video memory.

B. Large-scale Incremental Mapping

We demonstrate the unbounded reconstruction capability of RIM for large-scale scenes using the sequence 00 of the KITTI dataset [28], which provides real-world LiDAR data with a distance of approximately 3.7 kilometers. RIM outputs a complete and smooth map as shown in Fig. 6. Fig. 7 shows the RIM's total software memory usage in different devices. RIM can perform incremental mapping for extremely large outdoor scenes while avoiding exceeding video memory usage at the expanse of system memory usage thanks to the flexible map structure (Sec. II-B.1).

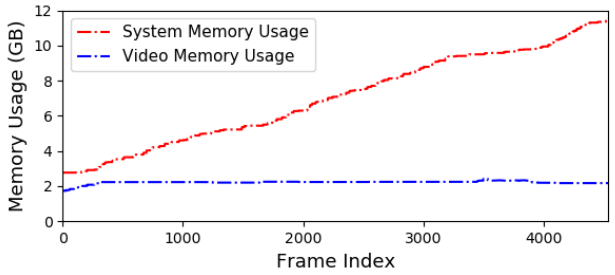


Fig. 7. Memory usage of RIM on different devices. The video memory usage remains constant at the expense of the system memory usage.

C. Runtime Analysis

We evaluate the average integration time per frame on the MaiCity and KITTI datasets, as shown in Tab. III. SHINE-Mapping fails in reconstructing KITTI 00 due to the limited video memory of our experiment platform. RIM's default iteration number per frame is 50; when the iteration number is 10, RIM can process online and obtain competitive reconstruction quality, as shown in Tab. II(**e**). RIM's each module's spending times are shown in Tab. IV, wherein the inference process consists of encoding and forward processes. The outlier removal module runs every second instead of every frame and takes an average of 8.56ms. Our running efficiency and reconstruction quality are significantly better than SHINE-Mapping. It is mainly contributed by the robot-centric mapping structure (Sec. II-B.1) and bundle supervision (Sec. II-B.2).

TABLE III

AVERAGE INTEGRATION TIME (MS) PER FRAME ON THE MAICITY AND KITTI DATASET

Datasets	SHINE	Ours	Ours (online)
MaiCity 01	1914.50	394.07	85.05
KITTI 00	-	398.07	95.09

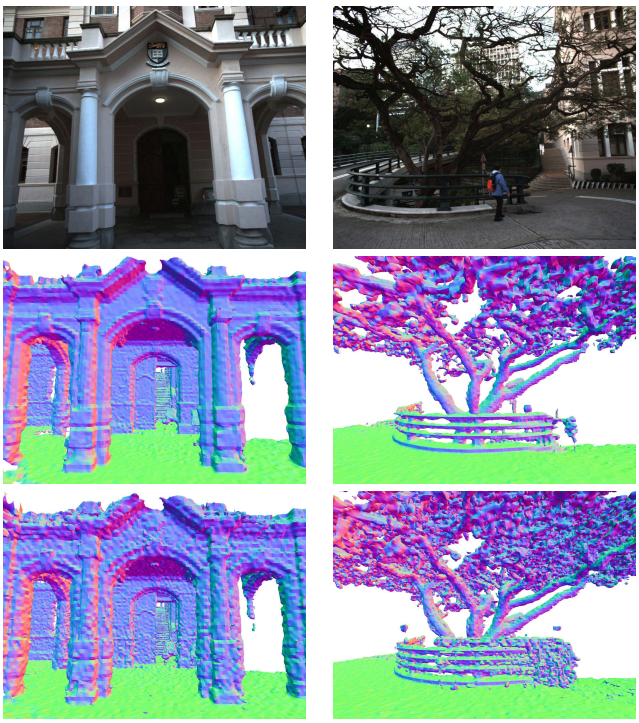
TABLE IV

AVERAGE TIME CONSUMPTION (MS) OF EACH MODULE PER FRAME

Sliding	PreProcess	**Per Iteration**			
		Sampling	Encoding	Forward	Backward
6.06	0.76	3.88	1.40	0.10	2.40

D. Mapping Application

This section presents a real-time incremental mapping application employing RIM with lidar inertial odometry [29]. It is assessed using the HKU Main Building dataset [30], which provides real-world lidar data sourced from a solid-state lidar RIM can mitigate the impact of noisy input with the help of outlier removal (Sec. II-B.2), producing commendable, detailed reconstruction results as depicted in Fig. 8. Noisy outliers are eliminated within the main building scene to yield a refined reconstruction. In situations involving trees and a traverse-walking pedestrian, our approach significantly reduces the influence of dynamic objects.



(a) HKU main building (b) Tree and walking person

Fig. 8. The reconstruction results on the HKU Main Building dataset, and the color represents the normal direction of the triangular surface. The top row displays the original scenes in color, the middle row shows the RIM reconstruction results with the outlier removal module, and the bottom row displays results without the outlier removal module.

IV. CONCLUSION

This paper introduced an efficient, scalable, robot-centric implicit mapping using range sensors. Key to our system is the subtle leverage of the robot-centric local map, which enhances both implicit model training efficiency and overall mapping quality while addressing the typical catastrophic forgetting phenomenon encountered in continual learning. We employ a flexible global map to preserve a reusable map. Our method, evaluated on various datasets, has proven to excel in terms of reconstruction quality, efficiency, and adaptability. Our method is tailored to mapping and depends on the precision of poses. Our future endeavors will include the incorporation of pose refinement during training and deploying our method into robotics tasks.

REFERENCES

[1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[2] K. Museth, "Vdb: High-resolution sparse volumes with dynamic topology," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, pp. 1–22, 2013.

[3] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[4] 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.

[5] I. Vizzo, T. Guadagnino, J. Behley, and C. Stachniss, "Vdbfusion: Flexible and efficient tsdf integration of range sensor data," *Sensors*, vol. 22, no. 3, p. 1296, 2022.

[6] 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.

[7] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[8] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3504–3515.

[9] 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.

[10] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[11] 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.

[12] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[13] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler, "Neural geometric level of detail: Real-time rendering with implicit 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 358–11 367.

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

[15] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations," *arXiv preprint arXiv:2210.02299*, 2022.

[16] D. Yan, J. Liu, F. Quan, H. Chen, and M. Fu, "Active implicit object reconstruction using uncertainty-guided next-best-view optimization," *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.

[17] P. Kaushik, A. Gain, A. Kortylewski, and A. Yuille, "Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping," *arXiv preprint arXiv:2102.11343*, 2021.

[18] 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.

[19] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[20] J. Liu, X. Li, Y. Liu, and H. Chen, "Rgb-d inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9573–9580, 2022.

[21] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. Ieee, 2011, pp. 127–136.

[22] Z. Wang, H. Chen, S. Zhang, and Y. Lou, "Active view planning for visual slam in outdoor environments based on continuous information modeling," *IEEE/ASME Transactions on Mechatronics*, 2023.

[23] Z. Wang, H. Chen, and M. Fu, "Whole-body motion planning and tracking of a mobile robot with a gimbal rgb-d camera for outdoor 3d exploration," *Journal of Field Robotics*, 2024.

[24] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.

[25] 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.

[26] 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.

[27] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss, "Poisson surface reconstruction for lidar odometry and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5624–5630.

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[29] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[30] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 672–10 678.