The following publication W. Guo, B. Wang and L. Chen, "NeuV-SLAM: Fast Neural Multiresolution Voxel Optimization for RGBD Dense SLAM," in IEEE Transactions on Multimedia, vol. 27, pp. 7546-7556, 2025 is available at https://doi.org/10.1109/TMM.2025.3599100.

NeuV-SLAM: Fast Neural Multiresolution Voxel Optimization for RGBD Dense SLAM

Wenzhi Guo, Bing Wang*, Lijun Chen*

***Abstract*—We introduce NeuV-SLAM, a novel dense simultaneous localization and mapping pipeline based on neural multiresolution voxels, characterized by ultra-fast convergence and incremental expansion capabilities. This pipeline utilizes RGBD images as input to construct multiresolution neural voxels, achieving rapid convergence while maintaining robust incremental scene reconstruction and camera tracking. Central to our methodology is to propose a novel implicit representation, termed *VDF* that combines the implementation of neural signed distance field (SDF) voxels with an SDF activation strategy. This approach entails the direct optimization of color features and SDF values anchored within the voxels, substantially enhancing the rate of scene convergence. To ensure the acquisition of clear edge delineation, SDF activation is designed, which maintains exemplary scene representation fidelity even under constraints of voxel resolution. Furthermore, in pursuit of advancing rapid incremental expansion with low computational overhead, we developed *hashMV*, a novel hash-based multiresolution voxel management structure. This architecture is complemented by a strategically designed voxel generation technique that synergizes with a two-dimensional scene prior. Our empirical evaluations, conducted on the Replica and ScanNet Datasets, substantiate NeuV-SLAM's exceptional efficacy in terms of convergence speed, tracking accuracy, scene reconstruction, and rendering quality.**

***Index Terms*—NeRF, SLAM, Neural Implicit Representation, Dense SLAM, Tracking, Mapping.**

I. Introduction

DENSE Simultaneous Localization and Mapping (SLAM) stands at the forefront of computational perception, offering the dual capabilities of estimating the pose of a camera and meticulously creating a detailed topographical representation of the surrounding environment. This technique has been smoothly integrated into many different applications, ranging from the automation of vehicular navigation to the nuanced intricacies of augmented reality experiences [1]–[4].

Conventional SLAM methodologies, predominantly based on explicit spatial representations such as point clouds [5]–[7], voxel grids [8], [9], and surfel [10], [11], often struggle with capturing detailed color and luminance, leading to inconsistent or sparsely populated maps. This undermines the precision and reliability of the mapping process, making it crucial to overcome these challenges. Neural Radiance Fields (NeRF) [12], utilizing deep neural networks, present a compelling solution by providing a compact yet powerful representation capable of rendering photorealistic scenes with minimal storage requirements. Initial explorations into NeRF-based SLAM methodologies [13], [14] have demonstrated their potential in providing detailed and continuous scene representations for enhancing dense RGBD SLAM frameworks.

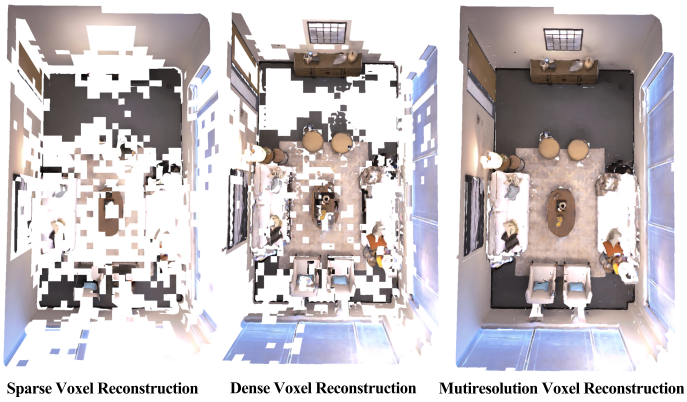

Fig. 1. We propose NeuV-SLAM, a SLAM system that incrementally reconstructs scenes from sequential RGBD frames. NeuV-SLAM reconstructs the scene separately based on dense voxels and sparse voxels.

The majority of existing typical implicit SLAM pipelines, as exemplified by [15], assume the spatial partitioning of scenes into grids [16] and depend on multi-layered grids and complex neural networks for scene representation, which limits their capability for scene expansion. Additionally, the complexity of these methods, coupled with their reliance on prior information or pre-trained decoders, hampers their ability to efficiently represent scenes. This significantly restricts their applicability in larger-scale and more complex scenes. Therefore, a question remains unanswered: **How can one scalably and efficiently learn the scenes through sequential RGBD frames**?

Implementing an incrementally scalable SLAM system in implicit environments presents significant difficulties. As RGBD image sequences are inputted, signifying the increment expansion of the scene, the SLAM system must be capable of real-time integration of new scene elements into the model while ensuring overall scene consistency and low computational resources. This necessitates an efficient management structure within the system that supports scene expansion. Recent research, as indicated in references [17], [18], has experimented with the use of voxel representations for scene depiction and management through octree structures to achieve expansion of unknown scenes. However, due to the hierarchical segmentation inherent in octree structures, they face

W. Guo is with the Department of Computer Science and Technology, Nanjing University, Nanjing, China, and the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: wenzhi.guo@connect.polyu.hk).

B. Wang is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China (e-mail: bingwang@polyu.edu.hk)

L. Chen is with the Department of Computer Science and Technology, Nanjing University, Nanjing, China (e-mail: chenlj@nju.edu.cn)

challenges of high structural complexity and excessive data volume, which have become primary obstacles.

Moreover, there remain significant challenges in achieving efficient learning of scenes. The primary issue lies in the limited capacity of neural networks. As the scene expands, the network often faces the problem of forgetting previously learned scene elements while learning new ones. This leads to difficulty in forming a consistent representation of the entire scene. Additionally, existing implicit SLAM systems often employ volume density methods to capture scene surfaces, but these methods are not always direct and efficient. Also, the use of a single-size voxel grid struggles to balance between rapid and accurate scene capture and the consumption of computational resources. For instance, the Vox-Fusion [17] method attempts to store neural features directly in voxel vertices to accelerate the optimization process of the scene, but this approach still relies on larger-scale neural networks for accurate scene representation. Additionally, Point-SLAM [19] achieves precise capture of scenes by using point clouds. However, this method struggles with fast processing when optimizing a large number of neural points and incurs substantial computational resource consumption.

To tackle all these challenges, we present NeuV-SLAM, a dense SLAM system based on neural multiresolution voxels, to achieve efficient incremental scene expansion, as shown in Figures 1 and 2. It consists of two major components: 1) We design an efficient hash-based multiresolution voxel management structure, termed ***hashMV***, which robustly supports rapid dynamic scene expansion. Leveraging the innovative structure, our system achieves efficient scene reconstruction and incremental expansion in unknown environments while maintaining a small memory footprint. 2) To achieve efficient scene convergence, we propose an innovative implicit scene representation method, named ***VDF***, that anchors color features and SDF values directly within multiresolution voxels and employs an SDF activation strategy to enhance the capability of capturing finer scene details. Consequently, we are able to utilize a convergence-friendly lightweight decoder to learn scene colors by minimizing depth, color, and SDF losses, thus alternately optimizing the decoder and camera poses during the tracking and mapping phases. Finally, we comprehensively evaluate our method and the experiments demonstrate its competitiveness in terms of convergence speed, reconstruction quality, tracking accuracy, and rendering performance.

To summarize, our main contributions are as follows:

- We present NeuV-SLAM, an innovative RGBD dense SLAM based on neural multiresolution voxels, designed to efficiently accomplish incremental scene reconstruction and camera pose estimation in unknown environments.
- We develop a novel multiresolution voxel management architecture, named ***hashMV***, which is based on 2D information and hash structures. This facilitates efficient and scalable management of scenes.
- We propose a novel implicit scene representation, named ***VDF***, incorporating neural SDF voxels and SDF activation. This method involves directly anchoring SDF within multiresolution voxels and employing the activation function, enhancing the convergence efficiency and augmenting the ability to accurately fit detailed scenes.
- We conduct evaluations of our method using the Replica and ScanNet Datasets. Our approach exhibits strong competitiveness in convergence speed, reconstruction, tracking, and rendering capabilities.

II. Related Work

A. Dense SLAM

Contemporary advancements in dense SLAM systems have bifurcated predominantly traditional and learning-based approaches. Traditional SLAM, highlighted in works like [20], [21], relies on complex algorithms for 3D scene reconstruction. Learning-based SLAM, discussed in [22]–[25], uses machine learning to improve scene reconstruction.

Traditional SLAM advancements include KinectFusion [26] for simple indoor mapping and ElasticFusion [27] for large-scale adaptability. Systems like DVO SLAM [28] and ORB-SLAM3 [7] offer precision in depth estimation and feature-rich areas, respectively, while InfiniTAM [29] and OpenLORIS-Scene [30] cater to diverse and dynamic environments.

Recent research integrates learning-based modules into SLAM, such as CNNs for stereo matching [31] and deep networks for end-to-end feature processing [32]. Techniques like odometry estimation also benefit from deep learning, merging visual and inertial data for motion estimation [33]. These advancements showcase the versatility of dense SLAM, though explicit representations have limitations in applicability.

B. Neural Implicit Representation

The quest to accurately represent three-dimensional scenes in computer vision constitutes a significant scholarly challenge. This field has seen substantial contributions, as evidenced in advanced works [34]–[36]. These methods include various implicit forms like signed distance fields [37], occupancy fields [38], and radiance fields [12], applicable in three-dimensional reconstruction, novel view synthesis, and 3D generative models [39], [40]. Particularly, Neural Radiance Fields (NeRF) [12] have spurred significant progress, demonstrating the potential of neural implicit representations in rendering detailed environments [41], [42].

However, the field faces challenges like long training and rendering times. To improve efficiency, various techniques have been explored, including advanced precomputation and optimization of NeRF's multi-layer perceptrons, despite potential memory trade-offs [43], [44]. Enhancing NeRF-SLAM system efficiency remains an ongoing challenge.

C. NeRF based SLAM System

In NeRF-based SLAM, leveraging Neural Radiance Fields enhances camera pose estimation and environmental mapping. Noteworthy is BARF [45], which merges NeRF parameter and camera pose optimization through adaptive position encoding, refining the registration process. iMAP [13] pioneers as the initial online dense SLAM model using NeRF, optimizing camera poses and scene representations for ongoing learning. NICE-SLAM [14] builds on iMAP with keyframe and structure improvements, boosting detail, speed, and accuracy in

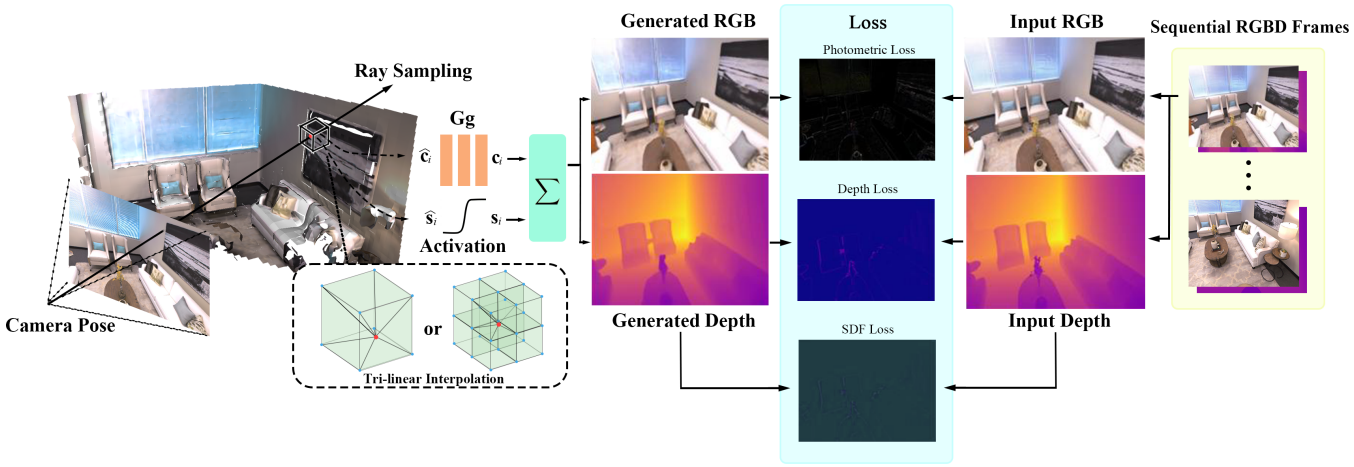

Fig. 2. **Overview of NeuV-SLAM**. NeuV-SLAM takes RGBD images as input and directly anchors color features and SDF values in multiresolution voxels to estimate camera pose and learn scene representation. From left to right, during the mapping stage, SDF values obtained directly through activated trilinear interpolation efficiently learn scene geometry, and scene color information is learned through interpolated neural features. The depth and color values are rendered through volumetric rendering, minimizing color, depth, and SDF losses to optimize the network G_g. From right to left, during the tracking stage, G_g parameters are fixed, and the camera pose is updated through backward propagation. Incremental expansion of the scene is achieved through the hashMV structure. The tracking and mapping stages alternate until the entire SLAM process is completed, with the multiresolution voxels converging to a finite set.

pose estimation. Its extension, NICER-SLAM [15], emerges as an advanced RGB-only SLAM system. Vox-Fusion [17] combines neural implicit representations with traditional fusion, employing voxel-based methods for scene optimization. Point-SLAM [19] introduces a point-based approach using monocular RGBD, focusing on tracking and mapping within a dynamic point cloud, optimizing for efficiency. The synergy of traditional and NeRF-based SLAM systems showcases the potential for advanced dense SLAM solutions [46], [47].

Our study critically examines traditional SLAM technologies, highlighting their limitations in scene reconstruction accuracy. To improve these traditional systems, we blend them with advanced learning methods to boost SLAM systems' adaptability and performance in complex environments. Our research specifically explores the use of neural implicit representations for mapping in dense SLAM tasks, suggesting that such advanced reconstruction techniques can lead to more accurate, efficient, and realistic maps. This enhances the functionality of autonomous systems in navigating intricate areas. We introduce a novel SLAM framework utilizing neural multiresolution voxels, designed to increase efficiency and scalability, making it suitable for challenging settings.

III. NeuV-SLAM

A. Multiresolution Voxel Generation and Management

1) Generation: Contrary to the [14], [17] approach, which employs a single-resolution voxel, we utilize multiresolution voxels based on scene details to represent the scene. This ensures efficient and accurate scene fitting with minimal computational cost. The efficacy of multi-resolution voxels is demonstrated in the ablation experiment F 2). In the mapping phase, subsequent to the successful tracking of a new frame, we execute an inverse projection of its depth estimation into a three-dimensional spatial domain. This is achieved by utilizing the pose of the current frame to translocate these tridimensional points into the global coordinate framework. Subsequently, these spatial points are annotated with edge attributes, derived from the edge information intrinsic to the current frame. The operational principle for voxel generation adheres to a 'first-come, first-served' protocol in relation to spatial point processing. Emphasizing the edge scene, a prioritization algorithm is employed for sorting these tridimensional points, preferentially processing those endowed with edge attributes. This methodology facilitates the generation of densely populated voxels, thereby mitigating the prospective overshadowing of dense voxels by their sparser counterparts. Following this, we compute a unique identifier (*Key*) for each point in the three-dimensional space, employing the subsequent methodology:

$$key = floor(\frac{p_{(x,y,z)}}{v}). \tag{1}$$

The *Key* serves as a hash index to identify the positional information of voxel, while $p_{(x,y,z)}$ represents the positional information of space points, and v signifies the edge length of voxels. This process involves querying the hash table: if the queried *Key* is already present in the table, it indicates a duplication of the corresponding spatial point, leading to its exclusion; conversely, if the *Key* is absent, a new voxel is allocated to the spatial point based on edge information. Within this framework, edge spatial points result in the generation of dense voxels, whereas regular spatial points produce sparse voxels. Notably, the resolution of dense voxels is twice that of sparse voxels. This design is to ensure proper alignment between the sparse and dense voxel grids and allows the grids to cover the entire scene without any overlap.

2) Management: Diverging from the approach in existing SLAM systems, which predominantly utilize a static single resolution and employ an axis-aligned voxel division approach for the entire scene [14], our method implements an innovative incremental scene expansion strategy, along with a multireso-

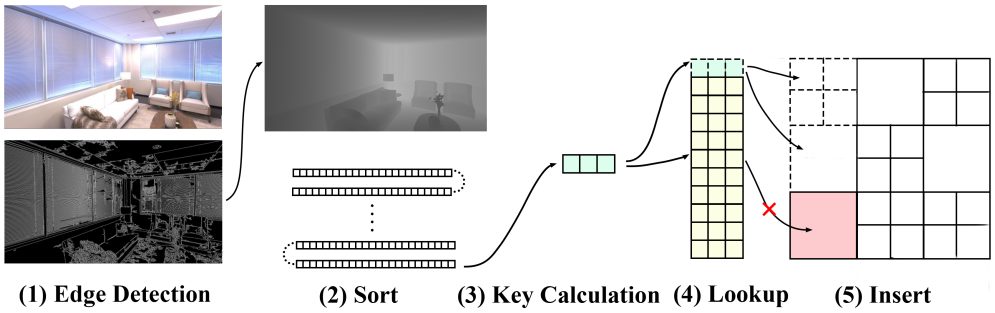

Fig. 3. **The process of multiresolution voxel generation. Edge Detection**: This step entails identifying boundaries within an image by pinpointing areas of significant brightness discontinuities. **Sort**: The detected edge points are then prioritized and resequenced, with a focus on these points for subsequent processing. **Key Calculation**: Utilizing the positional data of each point, a unique identifier or 'key' is computed. **Lookup**: This key is used to search within a hash table. Absent keys prompt new key generation. **Insert**: Voxel creation is guided by these keys. Edge points lead to denser voxel formation, while non-edge points result in sparser voxels. Existing occupied spaces result in key disposal.

lution voxel representation to preserve a greater level of detail information. Drawing inspiration from the design of traditional SLAM systems, we manage multiresolution voxels using a hash table, enabling incremental expansion in unexplored areas. We have named this structure, ***hashMV***; it facilitates the rapid dynamic addition of multiresolution voxels and the effective retrieval of adjacent voxels, significantly enhancing the efficiency of voxel generation and retrieval.

When integrating multiresolution voxels into our system, we assign each voxel vertex a unique index that is shared among adjacent voxels, a process complicated by the varying resolutions of voxels. To navigate this, we've developed a strategy for assigning indexes. When generating a new voxel, first ascertain the presence of a neighboring voxel. If the neighbor voxel is detected, derive the corresponding shared index based on the relative positions of the new and neighboring voxels, and then sequentially number the remaining vertices. In the absence of a neighboring voxel, directly sort and number the vertices in sequence. This approach of sharing identifiers conserves memory and notably accelerates the convergence of our neural voxel implementation.

B. Fast Neural Multiresolution Voxel Optimization

1) Multiresolution Points Sampling: In the pursuit of optimizing point sampling efficacy within our framework, our methodology entails the deployment of rapid ray tracing during the processing of each sampling ray. This is executed to ascertain the intersectionality of the ray with both sparse and dense voxel structures, thereby determining the occurrence of any interaction. In instances where a ray fails to intersect with the voxel ensemble, it is deduced that the corresponding pixel lacks contributory significance to the resultant rendering output, prompting its exclusion from the rendering pipeline. For rays that do exhibit intersections with voxels, uniform sampling is conducted along the trajectory between the ray-voxel intersection points, utilizing a predetermined step length denoted by S. This procedure yields sets of space points corresponding to the sparse and dense voxel domains. Finally, we merge the sampling points located on the same ray in both sparse and dense voxels and sort them by depth to facilitate subsequent volume rendering operations.

2) Neural Multiresolution Voxel Representation: Diverging from Vox-Fusion [17] and DVGO [48] methodologies, our voxel grid representation employs trilinear interpolation for simultaneous modeling of SDF and color features within voxel cells, which enhances precision in querying any space position, significantly boosting scene convergence efficiency:

$$\text{interp}\ (\boldsymbol{x}, \boldsymbol{V^{(D)}}, \boldsymbol{V^{(S)}}) : \boldsymbol{x} \in \mathbb{R}^3, \boldsymbol{V^{(D)}}, \boldsymbol{V^{(S)}} \in \mathbb{R}^{A \times N}. \tag{2}$$

x represents the 3D positional coordinates of the space point, V denotes the voxel grid, A is the dimension of the SDF or color feature, N is the number of dense or sparse voxels.

3) SDF Activation Strategy: The SDF voxel $V^{(SDF)}$ is utilized for storing SDF values in volumetric rendering. Within this framework, we use $\hat{s}$ to represent the original voxel SDF value before processing through the SDF activation. To enhance the representational capability of the SDF voxel grid while maintaining the intrinsic properties of SDF, we employ the hyperbolic tangent function (tanh) as the activation function to process SDF values $\hat{s}$, as follows:

$$s = \tanh \hat{s} = \frac{e^{\hat{s}} - e^{-\hat{s}}}{e^{\hat{s}} + e^{-\hat{s}}}. \tag{3}$$

The application of the hyperbolic tangent function (Tanh) enables the effective exploration of SDF values that are less than zero, concurrently facilitating the non-linearization of SDF values with the same sign. This methodology optimizes sensitivity to minute variations and enhances the model's precision in handling regions proximate to surfaces.

The interpolated values of the voxel SDF are subjected to a sequential processing regimen, involving the hyperbolic tangent function (tanh) and an interpolation function (interp). Taking inspiration from DVGO [48] and its post-activation strategy, the interpolation of voxel SDF values is sequentially processed using the interp and tanh functions for volume rendering. This method enhances the ability to produce well-defined, sharp surfaces, significantly improving the voxel grid's capacity to accurately capture and represent intricate geometric details. The formula of $s^{(post)}$ is as follows:

$$s^{(\text{post})} = \tanh\left(\text{interp}\left(\boldsymbol{x}, \boldsymbol{V}^{(\text{SDF})}\right)\right). \tag{4}$$

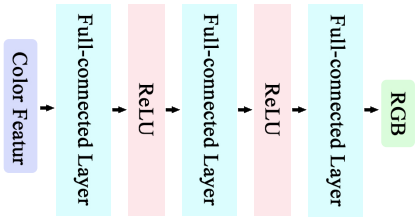

Fig. 4. The lightweight network architecture of G_g.

4) Volume Render: In contrast to existing methods such as Point-SLAM [19] and Vox-Fusion [17], which rely on neural networks to concurrently regress both identity and RGB values during their processing phases, this study introduces a more rapid and streamlined implicit representation approach, named ***VDF***. Our focus lies on the precise regression of RGB values for sampled points using a lightweight network designed for fast convergence, while directly optimizing the scene geometry through the neural SDF voxel grid.

Specifically, we perform a trilinear interpolation and activation strategy on the SDF dense voxel grid $V^{(D)(SDF)}$ and the SDF sparse voxel grid $V^{(S)(SDF)}$ based on the positional coordinates of the sampled points. That is, when the sampling point is located in a dense voxel grid, the interpolation is performed in the dense voxel grid, and the same is true for a sparse voxel grid. This process allows us to approximate and reconstruct the complex geometric structures of the scene with accuracy and continuity. Additionally, to model color, we introduce other voxel grids, $V^{(D)(RGB)}$ and $V^{(S)(RGB)}$. These grids are specifically designed to store and process color information, enabling precise modeling of scene color attributes. Consequently, we efficiently obtain the SDF and RGB values for any sampling point in the scene, as follows:

$$\begin{aligned} \hat{\boldsymbol{c}}_{\boldsymbol{i}} &= \operatorname{interp}\left(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{V}^{(\mathrm{S})(\mathrm{RGB})}\right) \vee \operatorname{interp}\left(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{V}^{(\mathrm{D})(\mathrm{RGB})}\right), \\ \hat{\boldsymbol{s}}_{\boldsymbol{i}} &= \operatorname{interp}\left(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{V}^{(\mathrm{S})(\mathrm{SDF})}\right) \vee \operatorname{interp}\left(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{V}^{(\mathrm{D})(\mathrm{SDF})}\right), \\ s_i &= \tanh \hat{s}_i. \end{aligned} \tag{5}$$

Subsequently, we employ a decoder, denoted as G, characterized by a set of tunable parameters represented as g, for the purpose of regressing RGB values. The architectural blueprint of this network is elucidated in Figure 4. During this procedural step, we utilize the color features, denoted $\hat{c}_i$, of the sample points as input, facilitating the acquisition of the respective color values associated with each sampled point. The equation of c_i is as follows:

$$c_i = G_g\left(\hat{c}_i\right). \tag{6}$$

Following this, we perform a holistic processing of the sampled points situated along the same ray, prioritizing them based on their depth, to ensure orderly processing:

$$r_j = \operatorname{sort}\left(r_j^D \cup r_j^S\right). \tag{7}$$

Then, we adopt a volumetric rendering technique similar to the Vox-Fusion [17] to compute and render depth D and color C attributes along the entire ray path. This process entails a meticulous examination of the gradient fluctuations within the SDF, facilitating the accurate identification of sampled points situated in close proximity to the neighboring regions of the scene's geometric surfaces. Additionally, we employ an effective strategy to exclude points that lie along the path between the scene surfaces and the camera sensor, thereby optimizing the efficiency of point sampling selection. The mathematical expression for this process is as follows:

$$\begin{aligned} o_i &= sigm\left(\frac{s_i}{st}\right) \cdot sigm\left(-\frac{s_i}{st}\right), \\ D_j &= \frac{\sum_{i=0}^{N-1} o_i \cdot d_i}{\sum_{i=0}^{N-1} o_i}, \\ C_j &= \frac{\sum_{i=0}^{N-1} o_i \cdot \mathbf{c}_i}{\sum_{i=0}^{N-1} o_i}. \end{aligned} \tag{8}$$

$sigm(.)$ represents the sigmoid function, where d_i stands for the depth of the sampled point, and st denotes the predefined SDF truncation distance, s_i is the predicted SDF value.

C. Tracking and Mapping

1) Tracking: In the tracking phase, we employed a replica of the implicit network and neural voxels created during the mapping phase, with their parameters fixed. To predict the current frame's camera pose, we utilized a basic zero-motion model, which estimates the six degrees of freedom (6-DoF) camera pose e $\in$ SE(3) starting from the known pose of the most recent frame. Subsequently, we refined the optimization of e$\in$SE(3) through a series of precise computational steps—including sampling of the scene, efficient volumetric rendering, and rigorous minimization of the rendering loss.

2) Mapping: In the mapping phase, our method involves the random selection of N rays for analysis within the current RGBD frame. The essence of this process lies in the precise minimization of color, depth, and SDF losses to guide the optimization of the implicit network and neural voxel features. To be specific, the color and depth losses are computed by contrasting the colors and depths acquired through volumetric rendering with the actual values observed along the rays in the scene. Simultaneously, the SDF loss quantifies the deviation between the adjusted predicted depth and the actual depth, taking into account the SDF values. S_p^{tr} represents the set of sampled points within the truncation distance. The mathematical expression for this loss function is as follows:

$$\begin{aligned} L = &\frac{1}{|R|}\sum_{i=0}^{|R|}\left\|\mathbf{C}_i - \mathbf{C}_i^g\right\| + \frac{1}{|R|}\sum_{i=0}^{|R|}\left\|D_i - D_i^g\right\| \\ &+ \frac{1}{|R|}\sum_{R\in p}\frac{1}{S_p^{st}}\sum_{s\in S_p^{tr}}\left(D_s - D_s^g\right)^2. \end{aligned} \tag{9}$$

This method goes beyond the consideration of color and depth consistency and, critically, provides a precise evaluation of the congruence between the geometric information and the scene. This rigorous approach guarantees a high level of accuracy and meticulous preservation of fine details in the scene reconstruction process.

In addition, we employ a mesh grid for scene reconstruction. Initially, a uniform sampling strategy is applied within both the sparse voxel and dense voxel spaces to generate a set of sampled points. Subsequently, while keeping the parameters of

the implicit network and neural voxels fixed, we determine the SDF values through the application of trilinear interpolation and activation strategy. Following this, the Marching Cubes algorithm [49] is utilized to process these SDF values, thereby ascertaining the positions of vertices and the corresponding facets, which are used to construct the mesh representation. Finally, the vertices and facets generated within the sparse voxel and dense voxel spaces are combined to form the geometric structure of the scene.

IV. Experiment

A. Experimental Setup

1) Datasets: Our study utilized the Replica and ScanNet Datasets for evaluation. The Replica Dataset, known for its 18 highly realistic indoor scenes, includes diverse settings like offices and kitchens. We tested on the same 8 scenes from this dataset as in the Vox-Fusion study. The ScanNet Dataset provided a wide range of real indoor RGBD data. We chose various scenes from ScanNet to represent the common challenges in real-world indoor spaces.

2) Baseline: For our comparative analysis, we employed iMAP [13], Vox-Fusion [17], NICE-SLAM [14], and DI-Fusion [50] as our baseline methods. These systems have been recognized for their exceptional capabilities in scene reconstruction and camera pose estimation, making them ideal references for evaluating the effectiveness of our approach.

3) Metrics: To evaluate our system, we used metrics for mesh reconstruction quality and camera tracking accuracy. Mesh accuracy (Acc.) measures the chamfer distance from our mesh to the ground truth, while Completion (Comp.) calculates the distance from the ground truth to our mesh. Camera tracking was assessed using the Absolute Trajectory Error (ATE). For rendering quality, we utilized the Peak Signal-to-Noise Ratio (PSNR), Multi-Scale Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). These metrics collectively provide a detailed evaluation of our scene reconstruction and camera pose estimation. We adopted the evaluation codes for tracking and mapping from Vox-Fusion [17], ensuring a consistent and thorough assessment.

4) Implemention Details: Our approach is implemented using PyTorch. To enhance the efficiency of execution, our methodology incorporates a multi-process paradigm, segregating the tasks of tracking and mapping. These processes are concurrently executed on a pair of GTX3090 graphics cards. For the voxel framework, we have chosen a voxel dimension of 0.2, coupled with a step increment of 0.1, facilitating precise sampling within the voxels at the junctures of ray interactions.

B. Tracking

In evaluating our system's tracking accuracy, we conducted tracking tests on the Replica and ScanNet Datasets, starting with a qualitative analysis by comparing our tracking trajectories to the actual trajectories from these datasets, as shown in Figure 5. This comparison indicated our system's superior tracking capabilities, especially in early tracking stages and under sudden trajectory changes, over Vox-Fusion. This improvement is credited to our advanced scene representation using multiresolution voxels. By embedding color and SDF values into the voxel grid, our system achieves quicker convergence, allowing for swift adaptation to and precise tracking of trajectory shifts, even in the initial training stages or amid significant viewpoint alterations. This capability ensures reliable and accurate tracking in various conditions, highlighting our method's efficiency in consistent trajectory maintenance in complex settings.

We conducted a quantitative comparison of our system with NICE-SLAM [14] and Vox-Fusion [17]. Table I presents the quantitative evaluation results on the Replica Dataset. These findings reveal that NeuV-SLAM demonstrates superior tracking performance compared to the current leading voxel-based methods. Our approach produces state-of-the-art results relative to these advanced voxel-based methods, further validating the efficacy of our method. As a real-world scene dataset, ScanNet poses greater challenges. Table II presents the tracking performance of NeuV-SLAM on the ScanNet Dataset, where it continues to demonstrate superior performance, outperforming competing methods.

Methods	Metric	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
iMAP	**RMSE[cm]↓**	70.05	4.53	2.20	2.32	1.74	4.87	5.84	2.62	18.34
	mean[cm]↓	5.89	3.95	1.95	1.65	1.55	3.19	5.48	2.15	16.03
	mediam[cm]↓	44.78	3.35	1.73	1.35	1.37	2.35	47.56	1.86	13.04
NICE-SLAM	**RMSE[cm]↓**	1.7	2.38	2.9	**0.92**	**0.87**	1.74	3.38	2.93	2.1
	mean[cm]↓	1.47	2.0	1.56	**0.8**	**0.78**	1.48	2.2	2.17	1.54
	mediam[cm]↓	1.3	1.8	1.15	0.7	**0.7**	1.29	1.55	1.69	1.27
Vox-Fusion	*RMSE[cm]↓*	*0.27*	*1.33*	*0.47*	*0.70*	*1.11*	*0.46*	*0.26*	*0.58*	*N/A*
	mean[cm]↓	*0.22*	*1.07*	*0.30*	*0.49*	*0.73*	*0.37*	*0.22*	*0.40*	*N/A*
	mediam[cm]↓	*0.20*	*0.78*	*0.27*	*0.29*	*0.45*	*0.31*	*0.19*	*0.28*	*N/A*
Vox-Fusion*	**RMSE[cm]↓**	**0.64**	1.13	**0.87**	10.47	1.07	**1.94**	1.14	**1.08**	2.17
	mean[cm]↓	**0.57**	0.93	**0.71**	6.43	0.92	1.84	1.06	**0.98**	1.57
	mediam[cm]↓	0.53	0.79	**0.63**	3.97	0.81	1.79	1.01	**0.92**	1.24
Ours	**RMSE[cm]↓**	0.83	**1.08**	0.89	1.48	1.99	1.99	**1.04**	1.23	**1.32**
	mean[cm]↓	0.70	**0.88**	0.77	0.92	1.81	**1.75**	**0.97**	1.08	**1.11**
	mediam[cm]↓	0.61	**0.73**	0.66	**0.69**	1.78	**1.46**	**0.94**	0.96	**0.98**

TABLE I
TRACKING RESULT ON THE REPLICA DATASET. THE DATA OF IMAP, VOX-FUSION ARE FROM [17], VOX-FUSION* AND NICE-SLAM ARE IMPLEMENTED BY THE OPEN-SOURCE CODE. WE CONDUCTED FIVE TESTS FOR EACH SCENE AND CALCULATED THE AVERAGE VALUES. OUR METHOD OUTPERFORMS ALL EXISTING APPROACHES, WITH THE BEST RESULTS PROMINENTLY HIGHLIGHTED AS THE **FIRST**.

Methods	0000	0059	0106	0169	0181	0207	Avg.
DI-Fusion	62.99	128.00	18.50	75.80	87.88	100.19	78.89
NICE-SLAM	12.00	14.00	**7.90**	10.90	13.40	6.20	10.70
Vox-Fusion	16.55	24.18	8.41	27.28	23.30	9.41	18.52
Ours	**12.71**	**9.70**	8.50	**8.92**	**12.72**	**5.61**	**9.68**

TABLE II
TRACKING RESULT ON THE SCANNET DATASET. THE DATA OF DI-FUSION, NICE-SLAM, AND VOX-FUSION ARE FROM [19].

Methods	Metric	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
iMAP	**Acc.[cm]↓**	3.58	3.69	4.68	5.87	3.71	4.81	4.27	4.83	4.43
	Comp.[cm]↓	5.06	4.87	5.51	6.11	5.26	5.65	5.45	6.59	5.56
	Comp.Ratio[<5cm%]↑	83.91	83.45	75.53	77.71	79.64	77.22	77.34	77.63	79.06
NICE-SLAM	**Acc.[cm]↓**	3.53	3.60	3.03	5.56	3.35	4.71	3.84	3.35	3.87
	Comp.[cm]↓	3.40	3.62	3.27	4.55	4.03	3.94	3.99	4.15	3.87
	Comp.Ratio[<5cm%]↑	86.05	80.75	87.23	79.34	82.13	80.35	80.55	82.88	82.41
Vox-Fusion	**Acc.[cm]↓**	2.55	2.25	4.26	2.49	3.68	4.78	4.15	3.35	3.44
	Comp.[cm]↓	2.82	2.36	2.67	**1.64**	**1.78**	3.02	3.11	3.36	2.60
	Comp.Ratio[<5cm%]↑	90.97	92.72	89.88	**95.40**	**93.91**	**89.06**	87.59	86.12	90.71
Ours	**Acc.[cm]↓**	**2.06**	**1.91**	**2.1**	**1.94**	2.17	**2.2**	**2.21**	**2.08**	**2.08**
	Comp.[cm]↓	**2.64**	**2.33**	**2.21**	1.75	2.28	**2.83**	**2.86**	**3.1**	**2.5**
	Comp.Ratio[<5cm%]↑	**92.26**	**93.07**	**92.57**	94.61	92.06	88.98	**88.65**	**88.06**	**91.28**

TABLE III
RECONSTRUCTION RESULT ON THE REPLICA DATASET. THE DATA OF IMAP, NICE-SLAM, AND VOX-FUSION ARE FROM [17]. OUR SYSTEM DEMONSTRATES THE MOST OUTSTANDING PERFORMANCE.

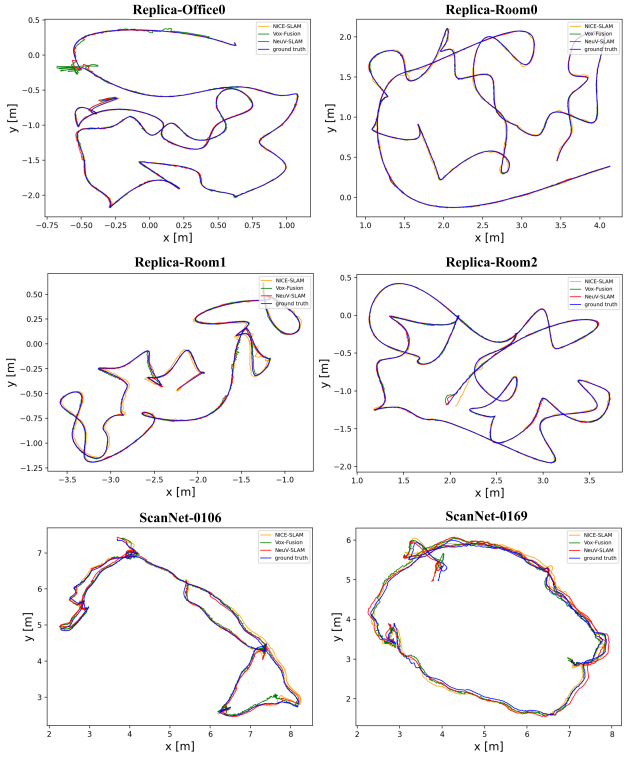

Fig. 5. Qualitative tracking result on the Replica and ScanNet Datasets. We project the trajectory in three-dimensional space onto the x-y plane.

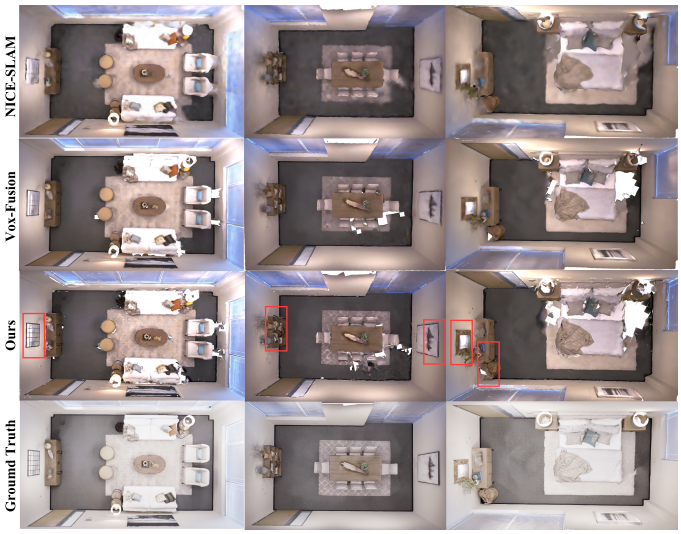

Fig. 6. Qualitative reconstruction results on the Replica Dataset. From top to bottom, we show the results of scene reconstruction of different methods (NICE-SLAM, Vox-Fusion, our method, and ground truth). To better visualize the differences in reconstruction between our method and other approaches, we employ red boxes to accentuate areas where our reconstruction method shows noticeable enhancements compared to other methods.

C. Mapping

In the qualitative evaluation against NICE-SLAM and Vox-Fusion, focusing on the Replica Dataset as depicted in Figure 6, we observed distinct characteristics of each system. NICE-SLAM, by assuming a complete surface across the entire observed space, tends to create surfaces in unobserved areas, which can lead to accurate reconstructions for minor gaps but significant deviations for larger ones. Vox-Fusion uses a single-resolution grid, limiting its ability to capture finer surface details. Our approach, similar to Vox-Fusion, models surfaces within visible sparse voxels, enabling realistic hole-filling. However, we further employ dense voxels for detailed texture capture, overcoming the single-resolution limitations. This technique enhances our system's ability to accurately represent and track complex surfaces, particularly valuable in environments with intricate surface details.

Table III quantitatively illustrates a comparative analysis of the existing iMAP, Vox-Fusion, NICE-SLAM, and our proposed method in the context of scene reconstruction on the Replica Dataset. The data for Vox-Fusion and NICE-SLAM are sourced from Vox-Fusion. Owing to the implementation of multiresolution voxel representation and the application of SDF activation mechanisms, our method demonstrates a significant advantage in the reconstruction of micro-level details.

D. Rendering

In Figure 7, the rendering capabilities of our NeuV-SLAM system are illustrated, demonstrating its proficiency in generating images with photo-realistic quality. Our approach stands out by rendering fine details more effectively than Vox-Fusion and NICE-SLAM, owing to its sophisticated multi-resolution voxel representation and neural processing techniques. This allows for a richer, more detailed visual output, capturing the nuances of the scene with greater fidelity.

Additionally, Table IV presents a quantitative analysis of rendering accuracy on the Replica Dataset. Compared to existing voxel-based advanced techniques, our method achieves state-of-the-art performance on the Replica Dataset.

Methods	Metric	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
NICE-SLAM	**PSNR[dB]↑**	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94	24.42
	SSIM↑	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856	0.809
	LPIPS↓	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198	0.233
Vox-Fusion	**PSNR[dB]↑**	22.39	22.36	23.92	27.79	29.83	20.33	23.47	25.21	24.41
	SSIM↑	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847	0.801
	LPIPS↓	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199	0.236
Ours	**PSNR[dB]↑**	**27.17**	**29.24**	**29.13**	**32.98**	**33.41**	**27.18**	**27.69**	**30.71**	**29.69**
	SSIM↑	**0.772**	**0.813**	**0.882**	**0.883**	**0.896**	**0.860**	**0.842**	**0.879**	**0.853**
	LPIPS↓	**0.223**	**0.197**	**0.211**	**0.208**	**0.147**	**0.214**	**0.189**	**0.157**	**0.193**

TABLE IV
RENDERING RESULT ON THE REPLICA DATASET. THE DATA OF NICE-SLAM AND VOX-FUSION ARE FROM [15].

E. Performance Analysis

1) Convergence Speed: To evaluate our system's convergence efficiency, we utilized two main strategies, starting with real-time scene rendering during the neural network's training phase. This allowed for a direct comparison of rendering quality at identical training data volumes and batch sizes, with rendering performed at each keyframe interval. This approach primarily facilitated a quantitative analysis of the network's convergence speed. Our method was closely compared to Vox-Fusion using the Replica Dataset, with findings detailed in Table V. The results showcased our method's superior image rendering quality given the same training data amount, underscoring our system's rapid convergence feature.

We also implemented a second evaluation technique, which entailed periodically rendering images from a consistent viewpoint throughout the training period. This approach aimed to monitor how the quality of the images evolved in relation

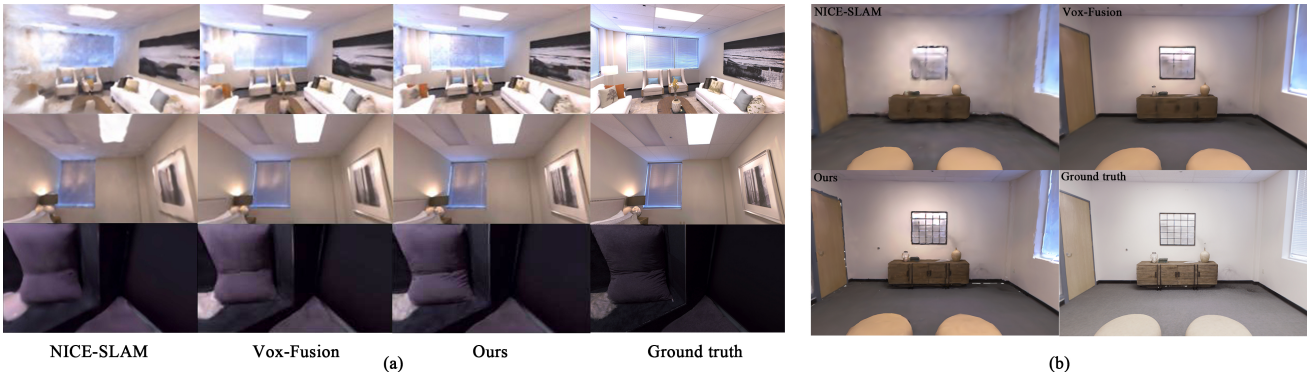

Fig. 7. (a) Qualitative rendering result in Replica Dataset. Our method exhibits superior rendering results in some details. (b) Qualitative rendering performance in scene mesh. The windows and cabinets in the mesh have better rendering details.

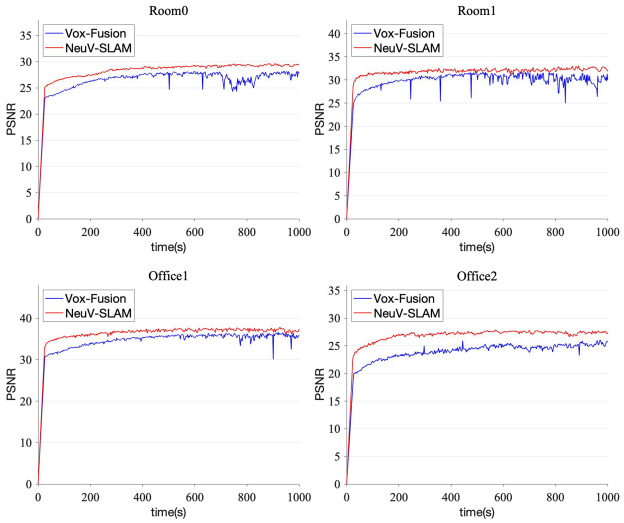

Fig. 8. Evaluation of the convergence speed between Vox-Fusion and our method. The horizontal axis is the system running time, and the vertical axis is the rendering PSNR value for the same viewpoint.

Methods	Ro0	Ro1	Ro2	Of0	Of1	Of2	Of3	Of4	Avg.
Vox-Fusion	26.06	28.29	28.61	29.93	32.28	26.65	26.32	29.37	28.43
Ours	**26.37**	**28.31**	**28.43**	**32.07**	**32.81**	**26.88**	**26.57**	**29.57**	**28.88**

TABLE V
RENDERING RESULTS UNDER THE SAME TRAINING VOLUME ON THE REPLICA DATASET.

to the length of the training process. The findings from this experiment revealed that our method not only reached convergence more swiftly than the Vox-Fusion technology but also produced images of superior rendering quality upon completion of training, as depicted in Figure 8. This demonstrates the efficiency and effectiveness of our approach in achieving high-quality results in a shorter timeframe.

2) Time and Memory Efficiency: Our study meticulously evaluated how multiresolution voxel-based sampling and rendering affect system runtime, focusing on the Replica Dataset. We measured the time needed for tracking and mapping in one iteration, detailed in Table VI. The results show our method, requiring separate sampling of sparse and dense voxels before merging, is slightly slower in the sampling phase than [17]'s single-step process. However, our approach's efficient scene convergence reduces total time by needing fewer iterations, maintaining performance on par with Vox-Fusion.

Methods	Tracking	Mapping
Vox-Fusion	*12ms*	*55ms*
Vox-Fusion*	25ms	22ms
ours	28ms	24ms

TABLE VI
THE TIME OF TRACKING AND MAPPING DURING A SINGLE ITERATION. THE DATA OF VOX-FUSION ARE FROM [17]. THE DATA OF VOX-FUSION* ARE IMPLEMENTED BY THE OPEN-SOURCE CODE.

The study also analyzed memory usage for the implicit scene decoder and voxel embeddings, comparing our NeuV-SLAM method with Vox-Fusion and NICE-SLAM in the Replica office-0 scene. Table VII shows that NeuV-SLAM has higher memory consumption for voxel embeddings than Vox-Fusion, attributed to storing SDF values in voxels. NICE-SLAM, with its four-layer voxel structure, has the highest memory usage among the compared methods.

Methods	Decoder	Embedding
NICE-SLAM	0.22MB	238.88MB
Vox-Fusion	*1.04MB*	*0.149MB*
Vox-Fusion*	0.43MB	1.22MB
ours	0.15MB	1.29MB

TABLE VII
THE MEMORY CONSUMPTION ON THE REPLICA DATASET. THE DATA OF NICE-SLAM AND VOX-FUSION ARE FROM [17]. THE DATA OF VOX-FUSION* ARE IMPLEMENTED BY THE OPEN-SOURCE CODE.

It is noteworthy that the memory footprint of our decoder part is less. This is because our decoder is only used for decoding color features, resulting in a more streamlined network structure—our network consists of three layers, while Vox-Fusion employs a six-layer structure. Consequently, our approach not only significantly reduces memory usage but also achieves higher reconstruction accuracy.

F. Ablation Study

1) Effectiveness of SDF Activation: Our study tested the SDF activation strategy's impact on tracking and mapping

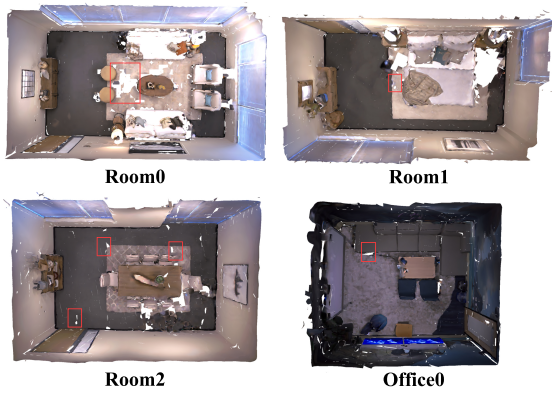

Fig. 9. Qualitative reconstruction result without SDF activation.

performance by conducting ablation experiments, removing the activation function to assess its role. Qualitative analysis, as shown in Figure 9, revealed that without the SDF activation, the system struggled with scene fitting, leading to gaps. The SDF activation function enhances scene fitting by introducing nonlinear estimation, thus improving reconstruction outcomes. Quantitative results in Table VIII contrasted with data in Tables I and III, further demonstrating the SDF activation strategy's benefits in camera pose tracking and scene reconstruction, notably by capturing finer details and speeding up scene convergence, which in turn benefits camera pose accuracy.

	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
Tracking	0.72	2.25	1.1	1.16	0.87	1.64	0.84	5.72	1.78
	0.58	1.32	0.87	0.83	2.39	1.40	1.48	5.04	1.74
	0.47	0.75	0.73	0.67	2.24	1.19	1.29	4.95	1.54
Reconstruction	2.73	2.99	2.79	2.36	2.94	2.61	2.72	2.52	2.73
	2.74	2.36	2.32	1.76	2.47	2.90	2.68	2.89	2.51
	90.25	91.81	90.52	93.90	90.56	87.65	89.64	87.16	90.31

TABLE VIII
TRACKING ANN RECONSTRUCTION RESULT ON THE REPLICA DATASET OF NEUV-SLAM WITHOUT SDF ACTIVATION.

2) Effectiveness of Multiresolution Voxel: In the ablation study contrasting single-resolution voxels with multiresolution voxels, we assessed tracking and mapping performance at voxel resolutions of 0.2 and 0.1, detailed in Table IX,X. Results showed superior performance at 0.1 resolution, highlighting resolution size's significant impact. When compared with multiresolution performance (Table I,III), the close performance parity confirmed the multiresolution strategy's effectiveness. Additionally, memory usage for the 0.1-resolution voxel in the Replica Room0 scene was 1.52 MB, higher than multiresolution voxels as shown in Table VII, demonstrating multiresolution voxels' advantage in reducing memory usage and facilitating large-scale applications.

V. CONCLUSION

In conclusion, we have explored the inherent limitations of traditional SLAM methods in accurately capturing detailed information and the difficulties associated with efficiently incrementally expanding scenes in NeRF-based SLAM systems. To address these challenges, we introduced NeuV-SLAM, a novel dense SLAM system based on neural multiresolution voxels. The key innovations of this research encompass the development of an efficient hash-based multiresolution voxel

	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
0.1 resolution	**0.66**	**0.83**	1.15	**1.18**	**1.98**	**1.76**	**0.95**	1.24	**1.20**
	0.58	**0.71**	0.91	**0.89**	1.83	**1.57**	**0.88**	1.16	**1.07**
	0.52	**0.63**	0.83	0.78	1.78	**1.33**	**0.86**	1.12	**0.95**
0.2 resolution	1.16	1.24	1.32	1.39	2.84	3.23	1.32	1.68	1.76
	1.01	1.12	1.14	1.07	2.62	2.85	1.26	1.52	1.69
	0.92	1.06	1.02	0.91	2.61	2.25	1.27	1.41	1.43
Muti-resolution	0.83	1.08	**0.89**	1.48	1.99	1.99	1.04	**1.23**	1.32
	0.70	0.88	**0.77**	0.92	**1.81**	1.75	0.97	**1.08**	1.11
	0.61	0.73	**0.66**	**0.69**	**1.78**	1.46	0.94	**0.96**	0.98

TABLE IX
TRACKING RESULT ON THE REPLICA DATASET IN DIFFERENT VOXEL RESOLUTION.

	Room0	Room1	Room2	Office0	Office1	Office2	Office3	Office4	Avg.
0.1 resolution	**1.56**	**1.33**	**1.55**	**1.90**	**1.22**	**1.67**	**1.79**	**1.74**	**1.59**
	3.05	2.50	2.40	**1.71**	**1.93**	3.19	3.11	3.51	2.67
	90.72	91.82	91.51	**94.88**	**93.30**	88.42	87.85	86.99	90.68
0.2 resolution	2.21	1.96	2.11	2.08	2.58	2.46	2.26	2.21	2.23
	3.08	2.56	2.50	2.01	2.58	3.69	3.23	3.35	2.87
	89.49	91.40	89.77	92.41	89.78	84.72	85.66	86.52	88.71
Muti-resolution	2.06	1.91	2.1	1.94	2.17	2.2	2.21	2.08	2.08
	2.64	**2.33**	**2.21**	1.75	2.28	**2.83**	**2.86**	**3.1**	**2.5**
	92.26	**93.07**	**92.57**	94.61	92.06	**88.98**	**88.65**	**88.06**	**91.28**

TABLE X
RECONSTRUCTION RESULT ON THE REPLICA DATASET IN DIFFERENT VOXEL RESOLUTION.

generation and management structure, hashMV, which facilitates rapid dynamic scene expansion while maintaining a compact memory footprint. Additionally, we propose a novel implicit representation method, anchoring neural features and SDF values directly within voxels, employing SDF activation strategy for efficient scene convergence and enhanced scene representation. Our method was rigorously evaluated on RGBD datasets, showcasing its competitive performance in terms of convergence speed, reconstruction quality, localization accuracy, and rendering capabilities. The progress made in enhancing SLAM technologies not only addresses the current limitations of SLAM methodologies but also opens up avenues for broader applications in fields such as robotics, autonomous navigation, and augmented reality.

ACKNOWLEDGEMENT

This work was jointly supported by the Young Scientists Fund of the National Natural Science Foundation of China (42301520), the Research Grants Council of Hong Kong (25206524), the Platform Project of Unmanned Autonomous Systems Research Centre (P0049516), the Seed Project of Smart Cities Research Institute (P0051028)

REFERENCES

[1] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2702–2719, 2019.

[2] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

[3] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3235–3242.

[4] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis *et al.*, "Bop: Benchmark for 6d object pose estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[8] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[9] 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.

[10] K. Wang, F. Gao, and S. Shen, "Real-time scalable dense surfel mapping," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6919–6925.

[11] J. Stückler and S. Behnke, "Multi-resolution surfel maps for efficient dense 3d modeling and tracking," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 137–147, 2014.

[12] 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.

[13] 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.

[14] 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.

[15] Z. Zhu, S. Peng, V. Larsson, Z. Cui, M. R. Oswald, A. Geiger, and M. Pollefeys, "Nicer-slam: Neural implicit scene encoding for rgb slam," *arXiv preprint arXiv:2302.03594*, 2023.

[16] C. Yan, D. Qu, D. Wang, D. Xu, Z. Wang, B. Zhao, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," *arXiv preprint arXiv:2311.11700*, 2023.

[17] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.

[18] Y. Mao, X. Yu, K. Wang, Y. Wang, R. Xiong, and Y. Liao, "Ngel-slam: Neural implicit representation-based global consistent low-latency slam system," *arXiv preprint arXiv:2311.09525*, 2023.

[19] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," *arXiv preprint arXiv:2304.04278*, 2023.

[20] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," in *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II 10*. Springer, 2008, pp. 802–815.

[21] B. Zhou, H. Mo, S. Tang, X. Zhang, and Q. Li, "Backpack lidar-based slam with multiple ground constraints for multistory indoor mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023.

[22] B. Zhou, C. Li, S. Chen, D. Xie, M. Yu, and Q. Li, "Asl-slam: A lidar slam with activity semantics-based loop closure," *IEEE Sensors Journal*, 2023.

[23] X. Yang, Z. Yuan, D. Zhu, C. Chi, K. Li, and C. Liao, "Robust and efficient rgb-d slam in dynamic environments," *IEEE Transactions on Multimedia*, vol. 23, pp. 4208–4219, 2020.

[24] H. Luo, Y. Gao, Y. Wu, C. Liao, X. Yang, and K.-T. Cheng, "Real-time dense monocular slam with online adapted depth prediction network," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 470–483, 2018.

[25] C. Chen, A. Jin, Z. Wang, Y. Zheng, B. Yang, J. Zhou, Y. Xu, and Z. Tu, "Sgsr-net: Structure semantics guided lidar super-resolution network for indoor lidar slam," *IEEE Transactions on Multimedia*, 2023.

[26] 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.

[27] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," in *Robotics: Science and Systems*, 2015.

[28] Z. Yan, M. Ye, and L. Ren, "Dense visual slam with probabilistic surfel map," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 11, pp. 2389–2398, 2017.

[29] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray, "Infinitam v3: A framework for large-scale 3d reconstruction with loop closure," *arXiv preprint arXiv:1708.00783*, 2017.

[30] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song *et al.*, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3139–3145.

[31] J. Zbontar, Y. LeCun *et al.*, "Stereo matching by training a convolutional neural network to compare image patches." *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2287–2318, 2016.

[32] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer, 2016, pp. 467–483.

[33] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[34] 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.

[35] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[36] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, "D-liom: Tightly-coupled direct lidar-inertial odometry and mapping," *IEEE Transactions on Multimedia*, 2022.

[37] 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.

[38] 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.

[39] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in neural information processing systems*, vol. 33, pp. 7462–7473, 2020.

[40] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 453–11 464.

[41] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.

[42] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 922–12 931.

[43] S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "Fastnerf: High-fidelity neural rendering at 200fps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 346–14 355.

[44] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "Autoint: Automatic feature interaction learning via self-attentive neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1161–1170.

[45] 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.

[46] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[47] C.-M. Chung, Y.-C. Tseng, Y.-C. Hsu, X.-Q. Shi, Y.-H. Hua, J.-F. Yeh, W.-C. Chen, Y.-T. Chen, and W. H. Hsu, "Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9400–9406.

[48] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5459–5469.

[49] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 347–353.

[50] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, "Di-fusion: Online implicit 3d reconstruction with deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8932–8941.

Wenzhi GUO received the B.E. degree from Taiyuan University of Technology, Taiyuan, China, in 2020, the M.S. degree from the University of Hong Kong, Hong Kong SAR, China, in 2021. He is currently working toward a dual Ph.D. degree from the Hong Kong Polytechnic University, Hong Kong SAR, China and Nanjing University, Nanjing, China. His research interests include 3D computer vision, SLAM, and robotics.

Bing WANG is an Assistant Professor in Robotics and Autonomous Systems at the Faculty of Engineering, The Hong Kong Polytechnic University. He obtained his DPhil degree in 2022 from the Department of Computer Science at the University of Oxford. His research is at the forefront of spatial intelligence, a dynamic field focused on advancing human-level 3D spatial perception and world understanding for mobile robotics. The primary objective is to enhance the reliability, intelligence, and security of intelligent machines in real-world environments.

Lijun CHEN , Professor at the National Key Laboratory of Computer Software New Technology at Nanjing University, Ph.D. supervisor, and Director of the Intelligent Robotics Research Institute at Nanjing University. He has been dedicated to research in the fields of intelligent robotics, IoT perception, computer vision, and artificial intelligence. With a focus on original innovation, he successfully developed the "Intelligent Supplies Inventory Robot," achieving a breakthrough from "0 to 1" and solving the global pain points of large-scale inventory being inaccurate, and slow. This contribution has provided a "Chinese solution" and "Chinese wisdom" to the long-standing challenge of large-scale material inventory. The "Intelligent Supplies Inventory Robot" has won the Special Gold Medal at the 46th Geneva's Invention Expo, the Grand Prize at the 22nd China International Industry Fair in 2020, and was selected as one of the ICIM Ten World Scientific and Technological Developments of Intelligent Manufacturing 2022.