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

VPE-SLAM: Neural Implicit Voxel-permutohedral Encoding for SLAM

Zhiyao Zhang[1], Yunzhou Zhang[1*], You Shen[1], Lei Rong[1], Sizhan Wang[1], Xin Ouyang[1], Yulong Li[1]

***Abstract*— NeRF can reconstruct incredibly realistic environmental maps in dense simultaneous localization and mapping, providing robots with more comprehensive scene map information. However, NeRF often struggles with geometric distortions in indoor reconstructions. To correct geometric distortions, we develop VPE-SLAM, based on the proposed voxel-permutohedral encoding, which can incrementally reconstruct maps of unknown scenes. Specifically, voxel-permutohedral encoding combines a sparse voxel feature grid created by an octree and multi-resolution permutohedral tetrahedral feature grids to represent the scene effectively. Especially when dealing with object edges, our method can effectively encode the geometry and texture of edges by the hybrid structural grid. We propose a novel local bundle adjustment module that utilizes a sliding window mechanism to manage adjacent keyframes requiring optimization. Furthermore, the proposed method establishes local map consistency by repeatedly optimizing keyframes that were initially under-optimized through a compensation strategy. The consistency of the local map can enhance the adaptability of our method to challenging scenes. Extensive experiments demonstrate that our method can achieve accurate camera tracking and produce high-quality reconstruction results on the Replica and ScanNet datasets. The source code will be available at https://github.com/NeuCV-IRMI/VPE-SLAM.**

I. Introduction

Dense simultaneous localization and mapping (SLAM) is a fundamental problem in 3D computer vision. The traditional dense SLAM can produce robust camera tracking and dense point cloud maps. However, traditional methods often generate numerous surface holes during reconstruction. Neural implicit representations utilize volume rendering of NeRF [1] and smooth priors from multi-layer perceptron (MLP) to reconstruct coherent scene geometry and overcome surface holes and map resolution limitations in traditional methods.

Neural implicit representations leverage MLP to learn the properties of sampled points within a scene. Among these works, NeRF can learn the color and volume density of 3D points in a scene from 2D images using MLP. SDF-based neural implicit methods [2]–[6] incorporate the signed distance function (SDF) into volume rendering of NeRF, allowing for surface geometry reconstruction. Therefore, NeRF can be used for dense reconstruction in SLAM, but its finite MLP capacity makes it unsuitable for large-scale scenes. NICE-SLAM [7] achieves scene feature storage through multi-level feature grids. The feature grids help prevent MLP from overwriting previously stored features but introduce significant computational complexity. Sparse parametric encoding reduces computational complexity by using hash tables and minimizing hash collisions with MLP. Concurrent methods [8]–[11] utilize sparsity on grid structures to enhance the optimization performance of NeRF. Co-SLAM [12] utilizes sparse parametric encoding for the efficient reconstruction of NeRF. However, the intensive correlations between the grid features can cause geometric distortions.

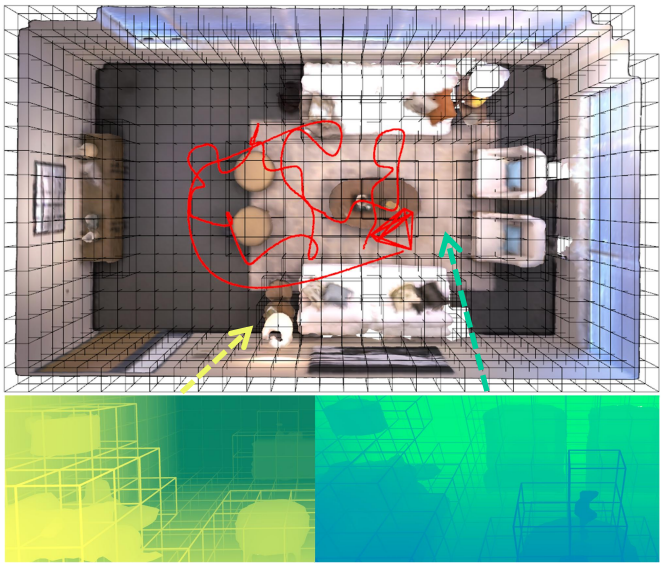


Fig. 1: **The demonstration of VPE-SLAM.** VPE-SLAM utilizes voxel-permutohedral encoding to reconstruct the scene and estimate camera poses (red line). The following two images are depth maps containing voxels for different scene parts.

The above raises a question: **How can we correct geometric distortions caused by NeRF to reconstruct accurate surface geometry in SLAM?** We present VPE-SLAM, which utilizes a voxel-permutohedral encoding (VPE) to correct geometric distortions, as shown in Fig.1. We use an octree to dynamically divide the unknown scene into sparse voxels and store geometry and color information at voxel vertices. This information of any point within voxels can be interpolated. Then, we incorporate tetrahedral feature grids into voxels to develop an enhanced spatial-aware VPE. VPE can effectively mitigate geometric distortions by the hybrid structural feature grid of voxels and tetrahedral. Additionally, we propose a novel local bundle adjustment (LBA) module with a sliding window mechanism to optimize adjacent keyframes continuously. Specifically, LBA adopts an under-optimized keyframe compensation strategy. When the window reaches its maximum capacity and a new keyframe is

*The corresponding author of this paper
[1]Zhiyao Zhang, Yunzhou Zhang, You Shen, Lei Rong, Sizhan Wang, Xin Ouyang and Yulong Li are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China. zhangyunzhou@mail.neu.edu.cn

This work was supported by National Natural Science Foundation of China (No. 61973066, 61471110) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049).

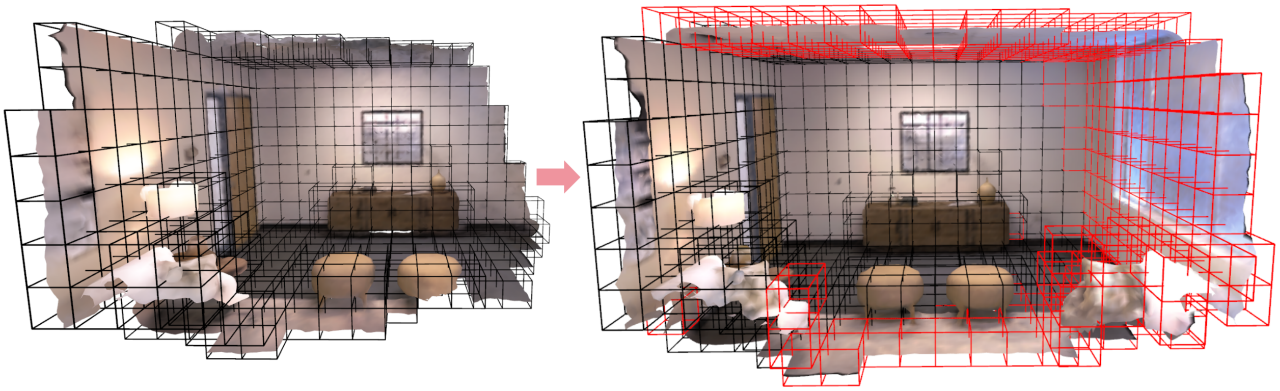


Fig. 2: **Dynamic voxel expansion in an unknown scene.** VPE-SLAM dynamically extends voxels using an octree based on the scene surface. Black cubes represent previously created voxels, while red cubes represent newly created voxels.

obtained, the compensation strategy retains keyframes with the highest pose uncertainty within the sliding window for more thorough optimization, thus establishing local map consistency. The experimental results demonstrate that VPE effectively mitigates geometric distortions, and LBA significantly improves the pose estimation accuracy.

The main contributions can be summarized as follows:

- We develop VPE-SLAM, which dynamically partitions surface in an unknown scene through voxel octrees and accurately reconstructs scene maps using our VPE with enhanced spatial awareness.
- We propose VPE, which effectively mitigates geometric distortions of NeRF by encoding scenes using a hybrid structural feature grid of voxels and tetrahedra.
- We design a sliding-window compensation optimization for LBA, enhancing the adaptability of VPE-SLAM to challenging scenes.
- Our proposed VPE-SLAM achieves accurate camera tracking and produces high-quality reconstruction results on the Replica and ScanNet datasets. We also conduct ablation experiments to demonstrate the effectiveness of VPE and LBA.

II. Related Works

A. Neural Implicit Representation

Recently, neural implicit representations for encoding scene geometry and color have gained popularity due to their strong expressiveness. NeRF variants [13]–[16] based on coordinate-based encoding can render stunning novel views. Concurrent approaches [17]–[21] combine various optimization methods and prior information to generate high-quality geometry. However, these approaches are challenging to optimize. To address this issue, sparse parametric encoding [22], [23] utilizes hash tables for rapid feature retrieval. However, it is prone to causing geometric distortions. Voxel-based representation [24]–[26] can recover sharp scene geometry using voxel structure but lacks fine-grained representation. Our proposed VPE, a hybrid structural feature grid composed of voxels and permutohedral tetrahedra, enables MLP to learn features with a heightened spatial context, thus alleviating geometric distortions.

B. Neural Implicit SLAM

Recent works have demonstrated that NeRF can be applied to SLAM, resulting in impressive geometric reconstruction results. iMAP [27] is the first to use NeRF in SLAM for predicting color and density in a scene without explicitly storing geometric shapes like point clouds. iMAP effectively addresses the issue of surface holes in traditional SLAM. To further enhance the geometric representation of the scene, NICE-SLAM [7] designs a multi-level hierarchical dense feature grid. However, it requires prior geometric knowledge of rooms to fill in unobserved viewpoints. Co-SLAM [12] adopts joint encoding based on coordinate-parametric, creating a coherent prior for the MLP to reconstruct a complete scene geometry. For camera tracking, Co-SLAM proposes a global bundle adjustment (BA) to optimize poses of overall keyframes but struggles with capturing local details. All these methods lack pose estimation accuracy, so we introduce LBA to compensate for under-optimized keyframes and enhance tracking precision. In addition, our approach directly renders geometry by automatically dividing scene surfaces via an octree and leverages VPE for fine-grained reconstruction.

III. Methodology

A. Approach Overview

VPE-SLAM adopts the pinhole camera model and assumes known intrinsic parameters $K \in \mathbb{R}^{3\times3}$, using a constant speed motion model. Then, VPE-SLAM performs the tracking and mapping processes to optimize the camera poses $T_i \in SE(3)$ and scene representation function F. And $\boldsymbol{\theta}$ is all the learnable functional parameters. Specifically, our proposed VPE and two MLP decoders constitute the scene representation function F. This function F maps world coordinates $\mathbf{x}$ into color $\mathbf{c}$ and SDF s.

$$F(\mathbf{x}, \theta) \mapsto (\mathbf{c}, s) \tag{1}$$

Most neural implicit SLAM methods require prior knowledge of scene dimensions and locations. To overcome this limitation, we dynamically partition the unknown scene with a voxel octree. As shown in Fig.2, whenever VPE-SLAM identifies a new keyframe, it projects 2D pixels into the 3D world. Sequentially, a voxel octree calculates morton codes based on the positions of 3D points and allocates voxels to

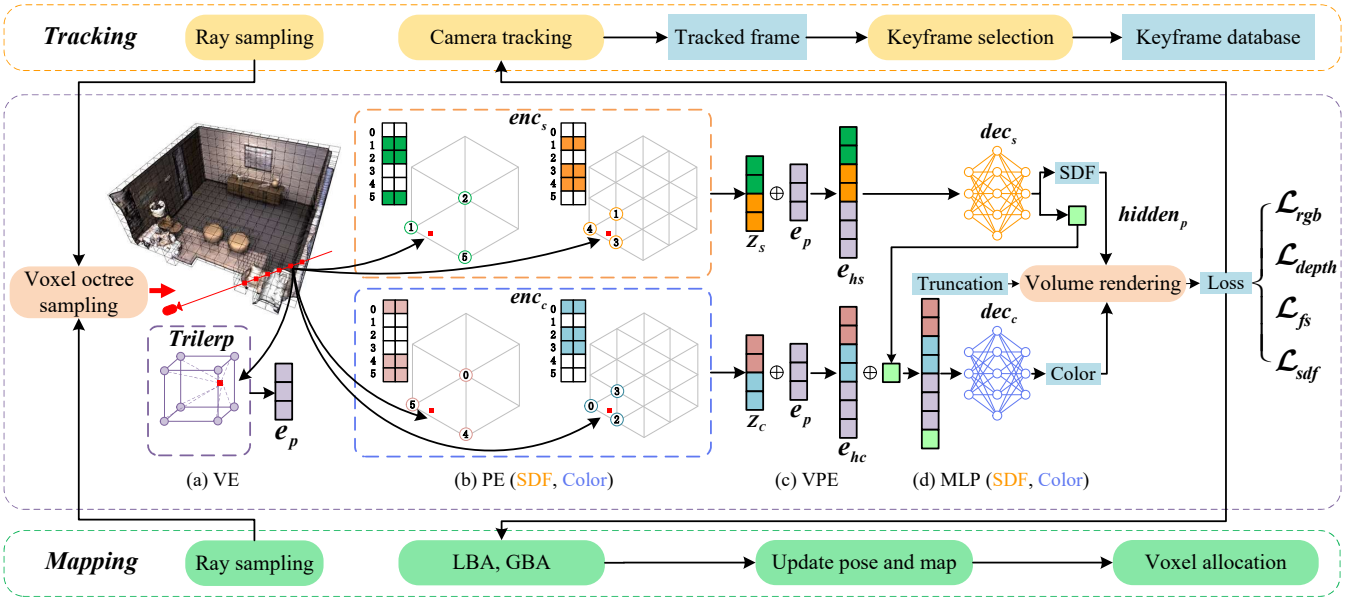


Fig. 3: **An overview of VPE-SLAM.** VPE-SLAM runs tracking and mapping. Whenever the tracking identifies a keyframe, VPE-SLAM runs mapping. When using voxel octree sampling, VPE-SLAM obtains 3D sampled points within voxels that the rays pass through. After that, it goes through four steps: (a) obtaining voxel-interpolated embedding e_p (VE) for sampled points, (b) obtaining permutohedral lattice encoding (PE) of SDF z_s and color z_c of sampled points. **For clarity, the diagram uses triangles instead of tetrahedra**, (c) separately concatenating e_p into z_s and z_c to obtain VPE of SDF e_{hs} and color e_{hc}, (d) using SDF decoder dec_s and color decoder dec_c to learn SDF and color. In training, we adopt TSDF-based volume rendering to optimize F in Eq.1. In mapping, VPE-SLAM uses LBA to further optimize the poses of local keyframes to establish local consistency. And then updates the map using global bundle adjustment (GBA).

enclose the surface of the scene. Inspired by Co-SLAM [12], VPE-SLAM optimizes more keyframes by sampling a small number of pixels from each keyframe in global BA (GBA). VPE-SLAM adopts a volume rendering based on TSDF [25] to obtain the final color and depth of the sampled pixels, as shown in Fig.3. The input RGB-D sequences are utilized as the supervisory signal during the training of our scene representation function F.

B. Voxel-permutohedral Encoding

As shown in Fig.3(a), every vertex of each cubical voxel is represented as a D-length feature vector, denoted as $e \in \mathbb{R}^D$. The eight vertices of the voxel v_i are denoted as Ω_i. For any 3D point $p \in \mathbb{R}^3$ within every voxel v_i, we can obtain its voxel embedding $e_p \in \mathbb{R}^D$ by voxel embeddings of eight voxel vertices using trilinear interpolation. The trilinear interpolation is denoted as the $Trilerp$.

$$Trilerp(e, \Omega_i, p) \mapsto e_p \tag{2}$$

Inspired by multi-resolution hash encoding [22], we adopt a multi-level grid structure and hash tables to store vertex feature vectors of tetrahedra. In contrast to the eight vertices of a hash grid cube, the four vertices of a tetrahedron can reduce the mutual influence of feature vectors between neighboring grids by half, significantly diminishing the correlation among vertices features. Moreover, the hybrid spatial structure incorporates features from different grid spaces into coordinates, allowing MLP to acquire enhanced spatial information and correct geometric distortions. As shown in Fig.3(b), we define a multi-resolution permutohedral tetrahedral grid for SDF enc_s and its output feature vector z_s and a multi-resolution permutohedral tetrahedral grid for color enc_c and its output feature vector z_c. Subsequently, as shown in Fig.3(c), the output e_p of trilinear interpolation is respectively concatenated with z_s and z_c to get the VPE feature of SDF e_{hs} and the VPE feature of color e_{hc}. We define θ_s as the parameters of enc_s and θ_c as the parameters of enc_c.

$$\begin{aligned} enc_s(p, \theta_s) &\mapsto z_s, \quad e_{hs} = e_p \oplus z_s \\ enc_c(p, \theta_c) &\mapsto z_c, \quad e_{hc} = e_p \oplus z_c \end{aligned} \tag{3}$$

Experiments demonstrate that VPE can alleviate distorted geometry and reconstruct fine-grained geometry and texture.

C. Decoder Architecture

VPE-SLAM represents the scene geometry as an implicit surface S, which corresponds to the zero-level set of SDF.

$$S = \{x \in \mathbb{R}^3 | SDF(x) = 0\} \tag{4}$$

As shown in Fig.3(d), we design two MLP decoders to decode SDF and color of sampled points within voxels intersected with each ray. VPE-SLAM adopts a single MLP decoder dec_s to learn SDF value $s_p \in \mathbb{R}$ of 3D point p. We define ϕ_s as the parameters of decoder dec_s. The L-length SDF hidden feature of the output is denoted as $hidden_p \in \mathbb{R}^L$. Similarly, we also define a single MLP decoder dec_c to learn color $\mathbf{c_p} \in \mathbb{R}^3$ of 3D point p. The parameters of decoder dec_c are denoted as ϕ_c. VPE-SLAM learns SDF and color of 3D sampled points through these two decoders, enabling the reconstruction of scene geometry and color.

$$\begin{aligned} dec_s(e_{hs}, \phi_s) &\mapsto (s_p, hidden_p) \\ dec_c(e_{hc}, hidden_p, \phi_c) &\mapsto \mathbf{c_p} \end{aligned} \tag{5}$$

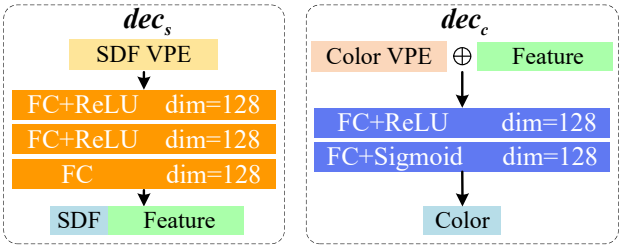


Fig. 4: **MLP decoder architecture.** The MLP decoders for learning SDF and color. A fully connected layer is denoted as FC.

As shown in Fig.4, the decoder for predicting SDF includes three fully connected (FC) layers of size 128. The decoder for predicting color includes two fully connected layers, each with a size of 128. Since surface color is determined by surface geometry, the learned feature from the predicted SDF is used as input to the color decoder dec_c.

D. Volume Rendering and Optimization

Given camera center $\mathbf{o}$ and direction $\mathbf{r}$, we uniformly sample M points using $\mathbf{x} = \mathbf{o} + d_p\mathbf{r}$ along each ray. d_p is the depth sampled along the ray. The sampled points within the voxels V intersected by the rays are considered valid sampled points p. Then, VPE-SLAM adopts a TSDF-based volume rendering to render sampled points p. Specifically, VPE-SLAM obtains the depth D, the color $\mathbf{C}$ of each ray by following the rendering function:

$$w_p = \sigma(\frac{s_p}{tr})\sigma(-\frac{s_p}{tr}), \quad \mathbf{C} = \frac{1}{|V|}\sum_{p\in V} w_p \cdot \mathbf{c_p}$$
$$D = \frac{1}{|V|}\sum_{p\in V} w_p \cdot d_p \tag{6}$$

where σ is the sigmoid function, color $\mathbf{c_p}$ and SDF s_p are predicted by dec_c and dec_s, and tr is a truncation distance. Then, VPE-SLAM adopts four loss functions of N rays as follows:

$$\mathcal{L}_{rgb} = \frac{1}{N}\sum_{i=0}^{N}\left\|\mathbf{C}_i - \mathbf{C}_i^{gt}\right\|$$
$$\mathcal{L}_{depth} = \frac{1}{|R|}\sum_{r\in R}\left\|D_r - D_r^{gt}\right\|$$
$$\mathcal{L}_{sdf} = \frac{1}{|R|}\sum_{r\in R}\frac{1}{S_r^{tr}}\sum_{x\in S_r^{tr}}\left(s_x - (D_r^{gt} - d_x)\right)^2 \tag{7}$$
$$\mathcal{L}_{fs} = \frac{1}{|R|}\sum_{r\in R}\frac{1}{S_r^{fs}}\sum_{x\in S_r^{fs}}(s_x - tr)^2$$

where R represents rays with valid depths, d_x is the depth sampled along the ray, color $\mathbf{C}_i^{gt}$ and depth D_r^{gt} are the color and depth ground truth values, $\mathbf{C}_i$ and D_r are obtained by Eq.6, S_r^{tr} is truncated surface region ($|D_r^{gt} - d_x| \leq tr$), S_r^{fs} is the region of the camera origin center to the positive truncated surface ($D_r^{gt} - d_x > tr$). $\mathcal{L}_{sdf}$ forces our function F to predict SDF s_x within the region S_r^{tr} close to the approximation of the SDF ground-truth $(D_r^{gt} - d_x)$. The free-space loss $\mathcal{L}_{fs}$ forces our function F to predict SDF s_x within the region S_r^{fs} close to truncation tr.

E. Local Bundle Adjustment

The tracking process primarily estimates the camera poses T_i and identifies keyframes. The mapping process relies on the global BA to perform optimization for keyframes. For keyframe selection, VPE-SLAM adopts a strategy of selecting keyframes at fixed intervals of Co-SLAM [12]. To compensate for the possibility of missing essential keyframes due to this selection strategy, we further develop a LBA to enhance local constraints and alleviate this phenomenon as much as possible. LBA adopts a classic sliding window mechanism, which consistently optimizes the previous few keyframes of the current frame. When the number of keyframes exceeds the maximum capacity of the window, LBA removes the frame with the lowest loss from optimization. This method ensures that those with slightly higher losses will be optimized as much as possible. Specifically, LBA places the seven most recent keyframes into a sliding window each time to establish a strict local keyframe constraint. Regarding implementation details, LBA runs only at the beginning of the mapping thread, aiming to create more accurate camera poses before the global BA. VPE-SLAM also performs joint optimization of camera poses and scene geometry during the LBA. Experimental results demonstrate that VPE-SLAM can achieve excellent tracking accuracy. In ablation experiments, we also validate the effectiveness of the LBA.

IV. Experiments

A. Experimental Setup

Datasets and evaluation. Our approach is evaluated on Replica [28] and ScanNet [29] datasets. Replica [28] is a synthetic dataset widely used in neural implicit SLAM research. ScanNet [29] is a more challenging real-world dataset with significant rotations and blurriness images. For pose estimation, we use Absolute Trajectory Error (ATE) to evaluate tracking accuracy on Replica [28] and ScanNet [29]. For evaluation of reconstruction quality, the accuracy (Acc.), completion (Comp.) and completion ratio (Comp. Ratio) are used on Replica [28].

Baselines. We use iMAP [27], NICE-SLAM [7] and Co-SLAM [12] as baselines. In terms of experimental details, iMAP is implemented from NICE-SLAM [7], although not the original implementation, it has been widely adopted in various works. We denote iMAP from NICE-SLAM [7] as iMAP*. Additionally, the number of iterations of tracking and mapping processes of Co-SLAM [12] is set to 10. For a fair comparison, we set the iteration count of Co-SLAM [12] to match ours. The version of Co-SLAM with increased iterations is denoted as Co-SLAM†.

Implementation details. We run VPE-SLAM on a desktop PC with an Intel Core i7-11700 (16 cores @ 2.50GHz), 32GB of RAM, and a single NVIDIA GeForce RTX 3090 GPU. VPE-SLAM uses voxels with a size of $0.2m$. Additionally, VPE-SLAM utilizes 18 hash tables, each with a size of 2^{19}. The truncation tr is $5cm$. The tracking process involves 20 iterations for every frame, and the mapping process involves 15 iterations.

TABLE I: **Quantitative results for camera tracking in the eight scenes of Replica [28].** We report the ATE RMSE↓, mean↓ and median↓ [cm] for camera tracking in the eight scenes of Replica [28]. Additionally, we calculate the average values of three metrics.

Method	Metric [cm]	Office0	Office1	Office2	Office3	Office4	Room0	Room1	Room2	Avg.
iMAP* [27]	RMSE↓	2.32	1.74	4.87	58.40	2.62	70.05	4.53	2.20	18.34
	mean↓	1.65	1.55	3.19	54.88	2.15	58.91	3.95	1.95	16.03
	median↓	1.35	1.37	2.35	47.56	1.86	44.78	3.35	1.73	13.04
NICE-SLAM [7]	RMSE↓	0.99	0.90	1.39	3.97	3.08	1.69	2.04	1.55	1.95
	mean↓	0.86	0.81	1.20	2.05	2.09	1.50	1.80	1.18	1.44
	median↓	0.76	0.74	1.09	1.28	1.53	1.38	1.67	0.98	1.18
Co-SLAM [12]	RMSE↓	0.55	0.49	1.99	1.37	0.80	0.66	1.24	1.20	1.04
	mean↓	0.46	0.43	1.82	1.29	0.69	0.57	0.80	0.88	0.87
	median↓	0.39	0.38	1.54	1.26	0.59	0.54	0.60	0.78	0.76
Co-SLAM† [12]	RMSE↓	0.45	0.46	1.88	1.23	0.68	0.45	0.63	0.78	0.82
	mean↓	0.37	0.43	1.71	1.17	0.58	0.42	0.56	0.73	0.75
	median↓	0.32	0.41	1.43	1.14	0.49	0.40	0.49	0.71	0.67
Ours	RMSE↓	**0.37**	**0.30**	**0.46**	**0.48**	**0.46**	**0.44**	**0.52**	**0.41**	**0.43**
	mean↓	**0.32**	**0.26**	**0.42**	**0.42**	**0.41**	**0.39**	**0.47**	**0.33**	**0.38**
	median↓	**0.30**	**0.24**	**0.39**	**0.40**	**0.38**	**0.37**	**0.43**	**0.29**	**0.35**

TABLE II: **Quantitative results for camera tracking in the six scenes of ScanNet [29].** We report the ATE RMSE↓ [cm] for camera tracking in the six scenes. Additionally, we calculate the average ATE RMSE↓ [cm] for each method.

Method	0000	0059	0106	0169	0181	0207	Avg.
iMAP* [27]	55.95	32.06	17.50	70.51	32.10	11.91	36.67
NICE-SLAM [7]	8.64	12.25	8.09	10.28	12.93	5.59	9.63
Co-SLAM [12]	7.18	12.29	9.57	6.62	13.43	7.13	9.37
Co-SLAM† [12]	**7.13**	11.14	9.36	**5.90**	**11.81**	7.14	8.75
Ours	9.24	**9.22**	**7.37**	6.06	14.51	**4.91**	**8.55**

TABLE III: **Quantitative results of the reconstruction quality on the eight scenes in Replica [28].** We report accuracy (Acc.) and completion (Comp.). Additionally, we also show the completion ratio (Comp. Ratio) calculated with a $5cm$ completion threshold.

Method	Acc. [cm] ↓	Comp. [cm] ↓	Comp. Ratio [%] ↑
iMAP* [27]	3.62	4.93	80.51
NICE-SLAM [7]	2.37	2.64	91.13
Co-SLAM [12]	2.10	2.08	93.44
Co-SLAM† [12]	1.95	**2.06**	93.57
Ours	**1.52**	2.14	**93.61**

B. Camera Tracking Evaluation

Replica dataset. For assessing pose estimation in Replica [28], we utilize ATE RMSE, mean and median as evaluation metrics. As shown in Tab.I, our approach outperforms iMAP [27] and NICE-SLAM [7]. iMAP lacks an effective scene representation, significantly increasing camera tracking errors. NICE-SLAM adopts dense grid representation, making it challenging to reconstruct local surface geometry, which affects pose estimation accuracy. The inadequate pose optimization limits the performance of Co-SLAM [12]. Hence, our approach outperforms it in camera tracking. Our approach achieves the highest camera tracking accuracy. Furthermore, the average of the three ATE metrics in our approach is all less than $0.5cm$.

ScanNet dataset. In the ScanNet [29] dataset, we compare the ATE RMSE between the camera poses estimated by our method and those from baselines. As shown in Tab.II, our approach exhibits the best performance in terms of average ATE RMSE. After leveraging VPE and optimizing poses with LBA, our approach surpasses the improved iterations of Co-SLAM† [12], ensuring stable tracking accuracy in highly rotating scenes. In certain scenes with rapid motion changes, it may lead to an insufficient number of optimizable voxels in the current view, which results in our method performing less effectively than Co-SLAM†. However, this issue can be addressed by adjusting the voxel size and voxel allocation strategy. Experimental results demonstrate that our method can adapt to severe rotations. Especially in scene-0207, our ATE RMSE is less than $5cm$.

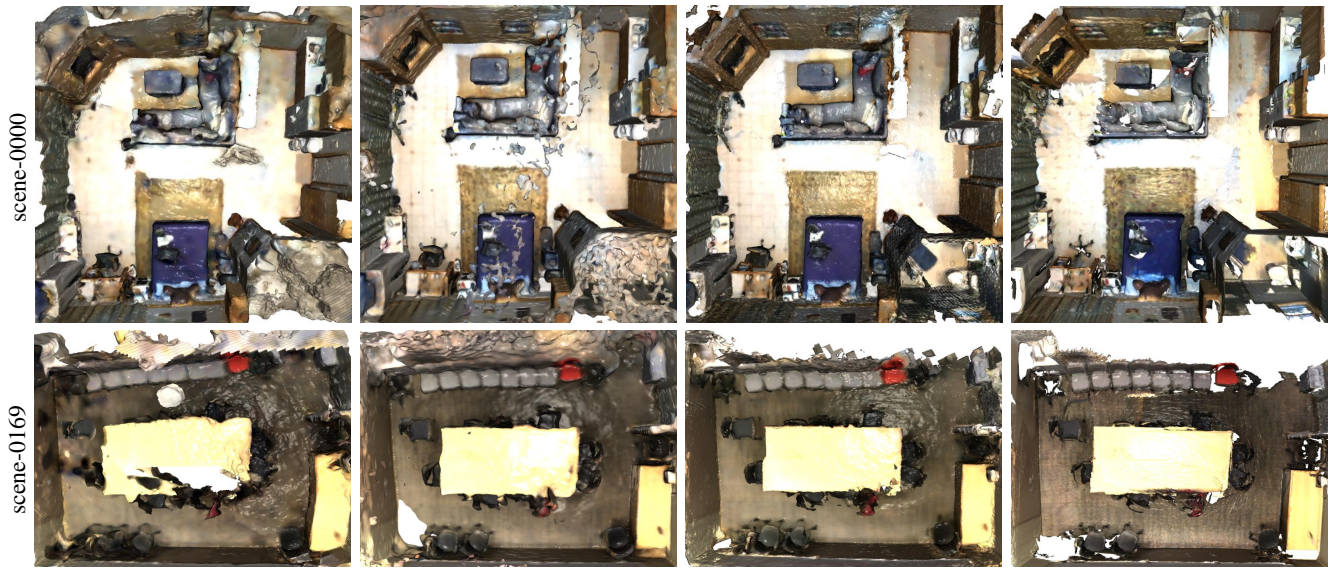
(a) Co-SLAM (b) Ours (c) GT

Fig. 5: **Alleviating geometric distortions by VPE.** VPE alleviates geometric distortions at the edges of objects, such as chairs.

C. Reconstruction Quality Evaluation

Alleviating local geometric distortions. The hybrid spatial structure of VPE provides a refined representation of local information near object edges. Therefore, VPE-SLAM can alleviate the geometric distortions of object edges. As shown in Fig.5, our approach not only reconstructs fine details but also corrects geometric distortions simultaneously, compared with Co-SLAM [12]. Additionally, our method outperforms Co-SLAM in rendering texture quality.

Reconstruction evaluation. For reconstruction quality evaluation, we evaluate our method on Replica [28] using the strategy proposed by Co-SLAM [12], which involves a virtual camera process strategy. As shown in Tab.III, our method outperforms iMAP* and NICE-SLAM [7] comprehensively. Co-SLAM samples all points within the scene, while our method samples only within the voxels, which may be slightly lower in some scenes. However, this issue can be addressed by adjusting the voxel size. The experi-



(a) NICE-SLAM (b) Co-SLAM (c) Ours (d) GT

Fig. 6: **The reconstruction results for ScanNet [29].** We report the reconstruction results of our method and other baselines in 0000 and 0169 of ScanNet, comparing them with ground truth (GT). Our approach accurately reconstructs the geometry and texture of the scene.

TABLE IV: **Quantitative tracking results for VPE ablation.** We report the average of ATE RMSE, mean and median [cm] for tracking of PE, VE and VPE in eight scenes of Replica [28].

	RMSE [cm]↓	mean [cm]↓	median [cm]↓
None	4.72	3.50	2.60
w/o PE	0.62	0.54	0.50
w/o VE	0.47	0.41	0.39
VPE	0.43	0.38	0.35

ments demonstrate that VPE-SLAM can reconstruct a more accurate implicit map by VPE and LBA compared to other methods. As shown in Fig.6, our method generates geometry closer to the ground truth mesh. The cluttered points from the roof obscure the bottom right corner of the scene-0000 result generated by Co-SLAM. However, our method effectively prevents generating some unnecessary points using voxels allocated by octree.

D. Ablation Studies

Effectiveness of VPE. We validate the effectiveness of VPE by camera tracking results of different encodings in eight scenes in Replica [28]. As shown in Tab.IV, VPE generates more accurate camera tracking results. In addition, to further validate the applicability of our VPE, we reduce the number of neurons in our two MLPs to 32 and adopt the fully-fused MLP [30] as the SDF and color of decoders. As shown in Fig.7, the voxel-tetrahedral structure of VPE enhances the spatial nature of encoding, resulting in more efficient parameter learning with MLP. Therefore, VPE can reconstruct more coherent geometry compared with PE.

Effectiveness of LBA. We validate the effectiveness of our LBA module on ScanNet [29] for camera tracking. We primarily use ATE RMSE as the evaluation metric. Tab.V shows that our method significantly improves after introducing LBA, especially in scene-0181. VPE-SLAM effectively improves the accuracy of pose estimation through the under-optimized keyframe compensation strategy of LBA.

(a) PE (b) VPE

Fig. 7: **Effectiveness of VPE.** We report the reconstruction results of (a) PE and (b) VPE using lightweight MLPs. VPE can generate coherent geometry compared with PE.

TABLE V: **Quantitative tracking results for LBA ablation.** We report the ATE RMSE↓ [cm] for camera tracking of our method without (**w/o**) and with (**w**) LBA in the six scenes of ScanNet [29].

	0000	0059	0106	0169	0181	0207
w/o LBA	12.23	11.19	7.86	6.22	FAILED	5.28
w/ LBA	9.24	9.22	7.37	6.06	14.51	4.91
Improvement	**2.99**	**1.97**	**0.49**	**0.16**	↑	**0.37**

V. Conclusion

This work focuses on an incremental and dynamic expanded neural implicit SLAM system that uses proposed voxel-permutohedral encoding (VPE) to alleviate geometric distortion. VPE utilizes a hybrid structural grid composed of voxels and permutohedral tetrahedra to achieve a compact representation of scene geometry, thereby alleviating geometric distortions in NeRF and enhancing the dense reconstruction performance of SLAM. In addition, we also propose a local bundle adjustment (LBA) module. LBA adopts a sliding window mechanism and an under-optimized keyframe compensation strategy to establish a consistent local map. The experimental results demonstrate that VPE-SLAM achieves accurate camera tracking and high-quality scene reconstruction on multiple datasets.

REFERENCES

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[2] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, volume 34, pages 27171–27183, 2021.

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

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

[5] Bowen Cai, Jinchi Huang, Rongfei Jia, Chengfei Lv, and Huan Fu. Neuda: Neural deformable anchor for high-fidelity implicit surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8476–8485, 2023.

[6] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.

[7] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12776–12786, 2022.

[8] Cheng Sun, Min Sun, and Hwann-Tzong 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*, pages 5459–5469, 2022.

[9] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *arXiv preprint arXiv:2206.05085*, 2022.

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[12] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023.

[13] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.

[14] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6260–6269, 2022.

[15] Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4287–4297, 2021.

[16] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. *Advances in Neural Information Processing Systems*, 34:24741–24752, 2021.

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

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

[19] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.

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

[21] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. In *2022 International Conference on 3D Vision (3DV)*, pages 433–442. IEEE, 2022.

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

[23] Radu Alexandru Rosu and Sven Behnke. Permutosdf: Fast multi-view reconstruction with implicit surfaces using permutohedral lattices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8475, 2023.

[24] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

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

[26] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.

[27] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021.

[28] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[29] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[30] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021.