iS-MAP: Neural Implicit Mapping and Positioning for Structural Environments

Haocheng Wang[1], Yanlong Cao[1]*, Yejun Shou[1], Lingfeng Shen[1], Xiaoyao Wei[1], Zhijie Xu[2], and Kai Ren[1]

[1] College of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China
00whcl@zju.edu.cn sdcaoyl@zju.edu.cn
[2] Xi'an Jiaotong-Liverpool University, Suzhou, 215123, China
Zhijie.Xu@xjtlu.edu.cn

Abstract. This work presents iS-MAP, a neural implicit RGB-D SLAM approach based on multi-scale hybrid representation in structural environments. iS-MAP encodes the scene using an efficient hybrid feature representation, which combines a 3D hash grid and multi-scale 2D feature planes. This hybrid representation is then decoded into TSDF and RGB values, leading to robust reconstruction and multilevel detail understanding. Additionally, we introduce Manhattan matching loss and structural consistency loss to fully incorporate the prior constraints of structured planes and lines. Compared with only color and depth losses, our structured losses are capable of guiding network optimization at the semantic level, resulting in more reasonable scene regularization. Experimental results on synthetic and real-world scene datasets demonstrate that our approach performs either better or competitive to existing neural implicit RGB-D SLAM methods in mapping and tracking accuracy, and predicts the most plausible reconstruction results for the unobserved structural regions. The source code will be released soon.

Keywords: Neural implicit mapping · Self localization · RGB-D SLAM · Structural constraints

1 Introduction

Simultaneous localization and mapping (SLAM) has long been studied as a fundamental problem in the field of computer vision and robotics, focusing on reconstructing the environment and self-localizing tasks. During the past two decades, visual SLAM has become more popular due to its low cost and ease of implementation, resulting in various sparse visual SLAM [3,13,20,21] and dense visual SLAM [22,23,38]. Despite the significant progress in tracking accuracy, these methods are mapping based on point cloud [13,20], cost volumes [23], surfels [28,38], or voxels [22] representations, which present serious challenges in achieving high-resolution and accurate reconstructions.

To achieve high-quality 3D scene reconstruction, emerging neural implicit representations, specifically represented by NeRF [18] have been employed for

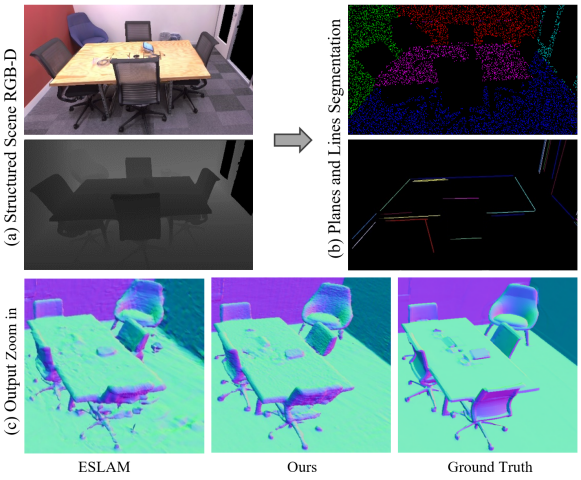

Fig. 1: We present iS-MAP, a neural RGB-D SLAM method for structured scenes. We propose a novel 2D-3D hybrid TSDF volume rendering technique with structured plane and line constraints. Our method shows reduced artifacts, high-fidelity reconstruction, and reasonable scene regularization

SLAM mapping, and neural implicit dense SLAM [11, 14, 26, 27, 31, 36, 37, 39, 42, 43] has been developed, allowing for reconstructing 3D surfaces with low memory consumption while maintaining stable camera tracking. These studies have primarily focused on man-made buildings, i.e., structural environments, which typically have numerous regularized plane and line features. While few efforts have incorporated structural regularities into the tracking and implicit mapping stages. Recently, Structerf-SLAM [36] has attempted to introduce plane features into neural implicit SLAM. However, its scene representation method struggles to ensure high-quality 3D reconstruction and overlooks the structural line feature in the environment.

In this study, we propose iS-MAP, a neural implicit dense SLAM approach with hybrid feature encoding and structural constraints. Our approach integrates 3D and 2D features using a hash grid and multi-scale plane fusion. Additionally, we introduce prior structural constraints in the mapping and tracking stages, respectively, to achieve more accurate localization and scene reconstruction. Fig. 1 illustrates an example of using structured plane and line constraints for improved scene reconstruction. In summary, the contributions of this article can be concluded as follows.

- We present a TSDF-based neural implicit SLAM approach that leverages a hash grid and multi-scale feature plane hybrid encoding, considering both 3D spatial features and 2D detailed features of the scene.
- In the mapping stage, structural consistency loss for plane and line regions has been employed to ensure the reconstructions are well-regularized and accurate.

– In the tracking stage, scene base planes are established according to the Manhattan hypothesis to align subsequent planes, leading to fast and stable data association and reduced drift.
– Extensive evaluations are conducted on both synthetic and real-world datasets to demonstrate iS-MAP method attains state-of-the-art reconstruction and camera tracking performance. Especially in unobserved structured regions, our method also achieves the most reasonable result.

2 Related Work

2.1 Dense Visual SLAM

Dense visual SLAM has been an interesting area for several decades due to its excellent ability for mapping. KinectFusion [22] utilized commercial RGB-D sensors to perform camera tracking by ICP and mapping via TSDF-Fusion. In contrast, ElasticFusion [38] adopts a map-centric approach, reconstructing a surf-based map of the environment. Further extensions include tracking pose optimization and loop closure [6, 23, 28]. Recently, methods incorporating deep learning have also shown outstanding performance, exhibiting superior accuracy and robustness compared to traditional methods. DeepV2D [32] employs neural networks to iteratively optimize depth and pose estimation. CodeSLAM [2] utilizes the optical flow definition for geometric residual calculation. DeepSLAM [15] employs an RCNN network for motion prediction. DROID-SLAM [33] employs a differentiable Dense Bundle Adjustment block for BA. However, these methods are still confronted with a significant challenge of high memory consumption when the resolution increases.

2.2 Structural Constraints

Structural constraints have often been leveraged in depth estimation [10, 35] and scene reconstruction [4, 8]. Our main focus is on its application in SLAM tasks. Point-plane SLAM [27] uses points and planes as primitives for registration. L-SLAM [12] uses the Manhattan Worlds hypothesis to further reduce rotation drift. Structure-SLAM [16] uses CNN to predict the normals of planes and lines to optimize the rotation of the shift and the reconstruction of the scene. Struct-SLAM [41] optimizes attitude estimation by adding line and plane structural features to an extended Kalman filter. PLP-SLAM [29] combinessemantic information from points and lines, performing a piecewise planar reconstruction (PPR). While these approaches introduce structural constraints, the scene representation and overall pipeline still follow the traditional SLAM approach.

2.3 Neural Implicit SLAM

Recently, neural implicit representation has shown outstanding potential in a variety of tasks, such as scene reconstruction [9, 25, 40] and new view synthesis

[1,17,18]. iMAP [31] is the first attempt to combine neural implicit representation with dense SLAM system, which employs an MLP for mapping and tracking. NICE-SLAM [43] utilizes a hierarchical layered feature grid to expand it to larger environments. Vox-Fusion [39] uses an octree-based approach to facilitate progressive mapping. MLM-SLAM [14] utilizes multi-MLPs to eliminate the need for pre-training of feature grids. ESLAM [11] employs feature planes on coarse and fine scales, which are more efficient than feature grids. Co-SLAM [37] uses joint encoding to ensure rapid convergence of the mapping process. SNI-SLAM [42] enables neural implicit semantic mapping. Point-SLAM [27] uses neural point clouds to represent scenes. However, none of these approaches leverage the structural constraints present in structural environments. Structerf-SLAM [36] utilizes feature grid representation and introduces planar constraints, but it stores each plane for matching, which reduces data association efficiency. It also neglects linear structural feature constraints in the environment. In comparison, our method, which employs a hash grid and multi-scale feature plane hybrid encoding, matches only the Manhattan base planes and maintains planar and linear consistency, achieving more efficient and accurate scene reconstruction.

3 Method

An overview of the proposed method is shown in Fig. 2, iS-MAP is a hierarchical hybrid encoding neural implicit SLAM system that incorporates structural constraints. The system enhances the 3D hash grid through multi-scale feature planes for scene encoding and introduces structural features to further strengthen prior constraints. The 3D-2D feature encoding with TSDF volume rendering is first introduced in Sec.3.1. Then, the planar and linear consistency and the Manhattan matching constraint in the structured scene are introduced in Sec.3.2. Finally, in Sec.3.3, we detail the method for global tracking and mapping.

3.1 3D-2D Hybrid Encoding with TSDF Volume Rendering

We propose a multi-scale 2D-3D hybrid feature coding for TSDF volume rendering. For 3D features, we utilize a hash grid [19] where the spatial resolution of each level is incrementally set between the coarsest R_{min} and the finest R_{max} resolution. At each sampling point x_i, we query the 3D features $\nu(x_i)$ using trilinear interpolation. For 2D features, we employ a three-directional multi-scale feature plane with four resolutions. Unlike the multi-dimensional features with only coarse and fine resolutions in [11], using fewer dimensions to extract scene features at multiple scales further enhances the perception of scene details and mitigates the hash collisions. On each feature plane, we query the plane features in three directions of the sampling points by linear interpolation. These features are concatenated to form the 2D features $\rho(x_i)$, which is challenging in feature grids due to significant memory consumption. The geometric decoder φ_g then accepts both 3D and 2D features, outputting the predicted TSDF value s_i and the TSDF feature vector h_i of x_i.

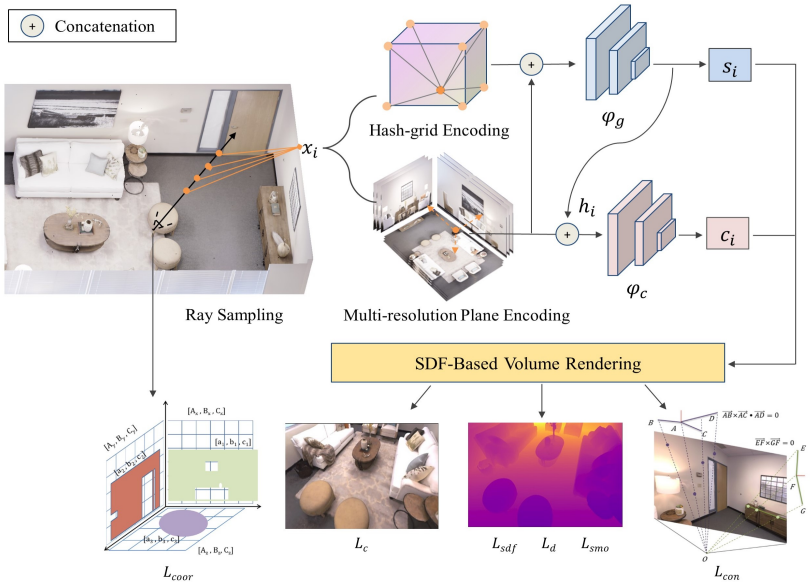

Fig. 2: Overview of the system: We sample the 3D points along the ray from each pixel and then encode the sample points by hybrid hash and multi-scale feature plane, and decode them to the TSDF value s_i and the TSDF feature h_i by the geometric decoder φ_g. Considering the consistency of geometry and appearance, h_i is then concatenated with the feature plane encoding $\rho(x_i)$ to predict the raw color c_i by appearance decoder φ_c. After TSDF volume rendering, the scene representation is optimized by minimizing sdf loss L_{sdf}, smooth loss L_{smo}, depth loss L_d , color loss L_c and structural consistency loss L_{con} in the mapping thread. Additionally, we also added Manhattan matching loss L_{coor} to the tracking thread to further optimize the camera pose.

$$(s_i, h_i) = \varphi_g(\nu(x_i), \rho(x_i)) \tag{1}$$

To improve color-geometric consistency, we concatenate the TSDF feature h_i and the multi-scale plane feature $\rho(x_i)$ to the color decoder φ_c and compute the color value c_i as Eq. (2).

$$c_i = \varphi_c(h_i, \rho(x_i)) \tag{2}$$

We render depth and color by integrating the predicted values along the sampled rays. Given the camera origin o and the ray direction r, we sample a total of N points, including N_{str} points along the ray and N_{sur} points near the surface. For each point x_i, we calculate their TSDF value s_i and color value c_i using Eq. (1) and Eq. (2). Subsequently, following the approach of [11, 24], we convert the TSDF value to volume density and calculate the termination probability w_i at each point of the ray.

$$\sigma(x_i) = \frac{1}{\beta}\text{Sigmoid}\left(\frac{s_i}{\beta}\right) \tag{3}$$

$$w_i = \exp\left(-\sum_{k=1}^{i-1}\sigma(x_k)\right)\left(1-\exp\left(-\sigma(x_i)\right)\right) \tag{4}$$

where β denotes a learnable parameter modulating the sharpness along the surface boundary.

Finally, for each ray, its depth and color can be rendered as Eq. (5).

$$\hat{D} = \sum_{i=1}^{N} w_i d_i \quad \hat{I} = \sum_{i=1}^{N} w_i c_i \tag{5}$$

3.2 Structural Prior Constraints

Segmentation of Line and Plane To integrate structural priors into our system, we select a keyframe every k frames and apply RANSAC [7] and LSD [34] methods to segment planes and lines with point counts exceeding a certain threshold, respectively. Fig. 1 shows an example of extracted plane and line segments. Notably, due to the typical sparse pixel sampling of volume rendering, we only generate sparse point clouds from the depth map and accelerate plane segmentation.

Structural Consistency Constraint Inspired by the self-supervised depth estimation [10,36], we introduced structural consistency loss in the plane region and line region respectively. Without specific regularization, volume rendering may not be able to maintain the flatness on plane regions in different views. Therefore, we apply a planar consistency loss to constrain these regions. For each plane, we randomly select four pixels and project them to 3D points A, B, C, and D using the render depth in Section 3.1.

The cross product of $\overrightarrow{AB}$ and $\overrightarrow{AC}$ should be orthogonal to the plane containing A , B , C and D. Thus, the dot product of $\overrightarrow{AB} \times \overrightarrow{AC}$ and $\overrightarrow{AD}$ should be equal to zero. The planar consistency loss L_{pc} can be expressed as Eq. (6).

$$L_{pc} = \frac{1}{N_p}\sum_{i=0}^{N_p}\left|\overrightarrow{A_iB_i} \times \overrightarrow{A_iC_i} \cdot \overrightarrow{A_iD_i}\right| \tag{6}$$

where N_p denotes the number of 4-point sets selected from plane regions randomly.

For the line regions, the linear consistency loss follows the strategy of planar consistency loss. Three pixels with their corresponding 3D points are randomly selected from a line region, denoted as E, F, and G. The cross product of vectors $\overrightarrow{EF} \times \overrightarrow{GF}$ should be a zero vector, making it a loss term, as shown in Fig. 3

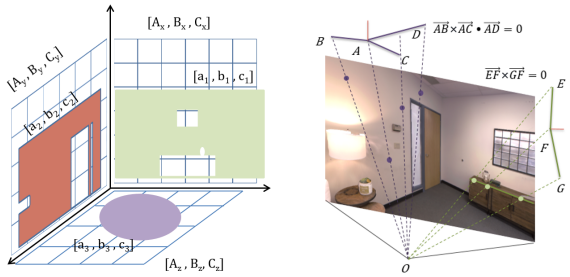

Fig. 3: Introduced structural prior constraints: Manhattan matching (left) and structural consistency (right).

(right). Similarly, we calculate the linear consistency loss L_{lc} using N_l 3-point sets from the line segments as Eq. (7).

$$L_{lc} = \frac{1}{N_l}\sum_{i=0}^{N_l}\left|\overrightarrow{E_iF_i}\times\overrightarrow{G_iF_i}\right| \tag{7}$$

Finally, the structural consistency loss L_{con} can be expressed as the weighted sum of L_{pc} and L_{lc}.

$$L_{\text{con}} = \lambda_{pc}L_{pc} + \lambda_{lc}L_{lc} \tag{8}$$

Manhattan Matching Constraint According to the Manhattan World (MW) hypothesis, planes in the structured scene are typically aligned with three primary directions. We parameterize the segmented planes using the Hessian form to apply this hypothesis, denoted as $\pi = [n, d]$, where, $n = (n_x, n_y, n_z)$ is the normal of the plane and d is the distance from the camera origin to the plane. Since the MW hypothesis is concerned only with the plane direction, we consider only the plane normal n in the Hessian form.

For the planes in the initial frame of each scene, we categorize them based on their normal directions. Planes with normal angles less than 10 degrees are classified. Subsequently, the plane with the most points in each category forms the base plane set of the scene. According to the MW hypothesis, subsequent planes should be parallel or vertical to these base planes. However, to avoid the influence of outliers and irregular planes, we only apply MW constraints to subsequently matched planes.

For each new frame, its planes are matched with the base planes. Specifically, for a plane π_c in the new frame, we project its normal n_c to the world coordinate system using the rotation matrix R_{cw} in the 6DoF pose, as Eq. (9).

$$n_c^{'} = R_{cw}^{-1}\cdot n_c \tag{9}$$

Then we check the angles between it and the base plane normals for matching. If the angle between the current plane π_c and a base plane π_i exceeds the

threshold θ_p, they are considered parallel and the parallel matching loss L_p can be calculated using the dot product of the normal vectors.

$$L_p = 1 - \overrightarrow{n_{wi}} \cdot \overrightarrow{n_c}' \tag{10}$$

where n_{wi} is the normal of parallel base plane π_i.

While if the normal angle between the current plane π_c and a base plane π_j is lower than the threshold θ_v, they are considered vertical and the vertical matching loss L_v can be calculated as Eq. (11).

$$L_v = \overrightarrow{n_{wj}} \cdot \overrightarrow{n_c}' \tag{11}$$

If the current plane doesn't match any parallel or vertical base plane, we don't process it. Fig. 3 (left) illustrates an example of the orientation alignment between the new frame's plane ($[a_x, b_x, c_x]$) and the base plane ($[A_x, B_x, C_x]$). Finally, the Manhattan matching loss L_{coor} is calculated as Eq. (12).

$$L_{coor} = \lambda_p L_p + \lambda_v L_v \tag{12}$$

3.3 Tracking and Mapping

Mapping When the first frame $\{I_0, D_0\}$ comes, we initialize our scene representation and create the base planes of the scene. For subsequent inputs, we update the scene representation on every keyframe. We first choose M pixels randomly from a sliding window of W keyframes. Next, we use the scene representation from Sec.3.1 to render and calculate the color and depth loss.

$$L_c = \frac{1}{M} \sum_{m=1}^{M} \left| I_m - \hat{I}_m \right| \tag{13}$$

$$L_d = \frac{1}{|R_d|} \sum_{r \in R_d} \left| D_m - \hat{D}_m \right| \tag{14}$$

where R_d is a set of rays with effective depth values in M pixels. Following [11], we also apply TSDF loss to every sampled point. Specifically, for sample points inside and outside the deep surface truncation region P_r^{tr} and P_r^{fs}, we use L_{tr} and L_{fs} respectively to calculate their TSDF loss, as Eq. (15) and Eq. (16).

$$L_{tr} = \frac{1}{|R_d|} \sum_{r \in R_d} \frac{1}{|P_r^{tr}|} \sum_{p \in P_r^{tr}} \left(d(p) + s(p) \cdot tr - D(r) \right)^2 \tag{15}$$

$$L_{fs} = \frac{1}{|R_d|} \sum_{r \in R_d} \frac{1}{\left| P_r^{fs} \right|} \sum_{p \in P_r^{fs}} \left(s(p) - 1 \right)^2 \tag{16}$$

where $d(p)$ represents the planar depth of point p to the camera, $s(p)$ is the predicted TSDF value, tr is the truncation distance, and $D(r)$ is the ray depth measured by the sensor. Notably, for points within the truncated region P_r^{tr}, we

set the importance of points where $|d(p) - D(r)| < 0.4tr$ to be four times higher than that of other points, thereby focusing more on points near the surface.

The final SDF loss is weighted by the two:

$$L_{sdf} = \lambda_{tr} L_{tr} + \lambda_{fs} L_{fs} \tag{17}$$

To reduce the noisy reconstructions caused by hash collisions in unobserved regions, we regularize the hash feature $\nu(x_i)$ by L_{smo} in a small random region in each iteration as [37].

$$L_{smo} = \frac{1}{|\eta|} \sum_{x \in \eta} \left(\Delta_x^2 + \Delta_y^2 + \Delta_z^2 \right) \tag{18}$$

where $\Delta_{x,y,z} = \nu(x_i + \varepsilon_{x,y,z}) - \nu(x_i)$ denotes the feature-metric difference between adjacent sampled vertices on the hash grid along the three dimensions.

The structural consistency loss L_{con} is utilized to further reinforce the structured prior constraints. Finally, we use Eq. (19) to jointly optimize the scene representation and keyframe poses as local bundle adjustment (BA).

$$\min \left(\lambda_d L_d + \lambda_{sdf} L_{sdf} + \lambda_{mc} L_c + \lambda_{smo} L_{smo} + L_{con} \right) \tag{19}$$

Tracking For camera tracking, we calculate the color loss, depth loss, and SDF loss using the same methods employed in the mapping thread. After obtaining the base planes of the first input frame $\{I_0, D_0\}$, we incorporate the Manhattan matching loss term L_{coor} every keyframe, as described in Sec.3.2. Ultimately, Eq. (20) is used for optimizing the current camera pose $[R|t]_j$.

$$\min \left(\lambda_d L_d + \lambda_{sdf} L_{sdf} + \lambda_{tc} L_c + L_{coor} \right) \tag{20}$$

4 Experiments

4.1 Experiments Details and Implementation

Dataset We evaluate the performance of iS-MAP on both synthetic and real-world sequences. (1) Replica [30], a synthetic dataset consisting of 8 high-quality room reconstructions. (2) ScanNet [5], a dataset collected from multiple sensors, containing challenging real-world RGB-D sequences.

Baselines and Metrics We use metrics from Co-SLAM [37] to measure the performance of our methods and existing state-of-the-art dense neural RGB-D SLAM methods [11,27,31,36,37,39,43]. For reconstruction quality, we use *Depth L1*(cm), *Accuracy*(cm), *Completion*(cm) and *Completion ratio*$(< 5cm\%)$. For tracking accuracy, we choose the commonly used $ATE \cdot RMSE$.

Implementation Details We run our system on a PC with a Intel Xeon Gold 5218R CPU and an NVIDIA Quadro P6000GPU. The hash grid resolution is 2 cm, and the hash map size is 16, resulting in a 36-dimension hash feature. For multi-scale feature planes, we vary the resolution 3 cm, 6 cm, 24 cm, and 48 cm from low to high. All feature planes have 3 channels in three inverse directions, yielding a 36-dimensional plane feature. The geometric decoders consist of two-layer MLPs with 64 channels in the hidden layer. The color decoders are also two-layer MLPs, with 64 and 32 channels in the hidden layers. We respectively select 256 and 128 sets of points for planar and linear consistency computation. For Replica, the weight of each loss are $\lambda_{tc} = 2$, $\lambda_{mc} = 4$, $\lambda_p = 0.2$, $\lambda_v = 0.02$, $\lambda_{pc} = 0.05$, $\lambda_{lc} = 0.05$ and $\lambda_{smo} = 0.01$. While for the ScanNet datase, we set $\lambda_{tc} = 5$, $\lambda_{mc} = 5$, $\lambda_p = 0.2$, $\lambda_v = 0.02$, $\lambda_{pc} = 0.05$, $\lambda_{lc} = 0.005$ and $\lambda_{smo} = 0.25$. We also keep the remaining hyperparameters consistent with ESLAM [11] as our reference.

Table 1: Comparison of the average reconstruction results for our method and other NeRF-based SLAM methods in Replica datasets. The best results were highlighted in red and the second best results were highlighted in blue. For the details of the evaluations for each scene, refer to the supplementary.

	Depth L1↓	Acc.↓	Comp.↓	Comp.Ratio↑	RMSE↓
iMAP [31]	4.64	3.62	4.93	80.50	2.58
NICE-SLAM [43]	1.90	2.37	2.63	91.13	1.95
Vox-Fusion [39]	2.91	**1.88**	2.56	90.94	1.03
Structerf-SLAM [36]	1.86	2.30	2.56	91.42	0.88
Co-SLAM [37]	1.51	2.10	2.08	93.44	0.86
ESLAM [11]	0.95	2.08	1.75	96.43	0.63
Ours	**0.75**	1.96	**1.66**	**96.64**	**0.48**

4.2 Results and Discussion

Replica We used the same rendered RGB-D sequence provided by iMAP [31] and conducted a quantitative evaluation of the results. As shown in Tab 1, our method achieves favorable results in all aspects except for the *Accuracy*, with a 21% improvement in *Depth L*1. Vox-Fusion [39] achieved the highest *Accuracy* by ignoring predictions in unobserved regions, resulting in almost the worst performance in other reconstruction metrics. Although our method only obtained suboptimal *Accuracy*, it made a better overall balance. We also select three sequences from the Replica dataset and visualize the reconstruction mesh in Fig. 4 qualitatively. The hybrid encoding of the hash grid and multi-scale feature plane enables our method to preserve finer details, such as plants in vases and the gap in cabinets. The structural constraints further regularize the reconstruction of planes and lines in space. For instance, areas like walls, floors,

Fig. 4: Reconstruction results of our method with the baseline on the Replica dataset [30]. We visualize three selected scenes and highlight details with colored boxes. Our method achieves higher precision in geometric details and more accurate structural features.

edges of tables and chairs exhibit overall flatter profiles with fewer artifacts and fluctuations. We also report the *RMSE* of camera tracking for the replica dataset in Tab. 1. Our method achieves the best results and up to 24% relative increase in tracking accuracy.

ScanNet We also benchmark iS-MAP and other methods on five randomly selected large scenes from ScanNet [5] to evaluate their scalability in real-world scenes. Due to the ScanNet dataset did not have complete ground truth meshes, we qualitatively analyzed the reconstruction results. As shown in Fig. 5. Our

Table 2: Camera tracking result on ScanNet(RMSE). The best results are in bold

Method	0000	0059	0106	0169	0207	Avg
iMAP* [31]	55.95	32.06	17.50	70.51	11.91	37.58
NICE-SLAM [43]	8.64	12.25	8.09	10.28	5.59	8.97
Vox-Fusion [39]	8.39	9.18	7.44	6.53	5.57	7.42
Structerf-SLAM [36]	7.28	**6.07**	8.50	7.35	7.28	7.30
Point-SLAM [27]	10.24	7.81	8.65	22.16	9.54	7.92
Co-SLAM [37]	7.18	12.29	9.57	6.62	7.13	8.12
ESLAM [11]	7.32	8.55	7.51	6.57	5.71	7.13
Ours	**6.45**	8.63	**7.32**	**5.85**	**4.61**	**6.57**

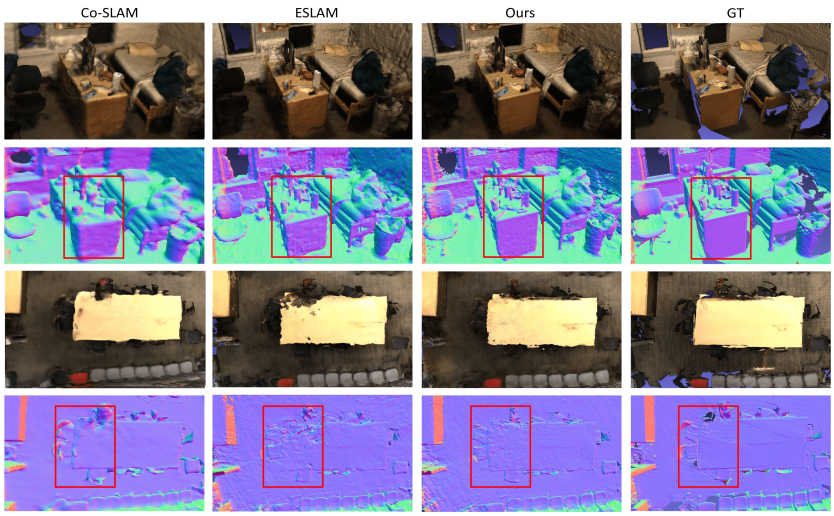

Fig. 5: Reconstruction results and corresponding normal maps of our method with the baseline on the ScanNet dataset [5]. To better show the difference, we use red boxes in the figures to indicate the improvements.

method can better restore the details of objects on the desk and provide a more square and complete desktop.

We quantitatively compared the camera tracking result in Tab. 2, our method has notably achieved the best results, exhibiting superior robustness. This is attributed to structural consistency prior constraints and stable Manhattan matching data association, which can reduce the drift issue during camera tracking.

Prediction for Unobserved Structural Regions One of the key strengths of neural implicit SLAM is its excellent predictive ability. For unobserved structural regions like walls, floors and ceilings, which are prevalent in real scenes, our approach benefits from prior regularization of structural consistency, enabling it to make the most reasonable predictions. As illustrated in Fig. 6, we selected reconstruction results from several typical scenes in Replica [30], in which the regions of the ceilings or walls have not been observed completely due to the camera angles, indicated by black holes in the figure. These regions reconstructed by our method are the flattest, aligning closely with the real scene.

4.3 Ablation Studies

Runtime and Complexity We evaluate the speed, memory and computational complexity of our method and others on office0 of Replica [30], as shown in Tab. 3. We report the tracking and mapping time for each frame, the scene representation and decoder size (#Param), and the number of floating-point operations (FLOPs) required for querying color and volume density of one 3D

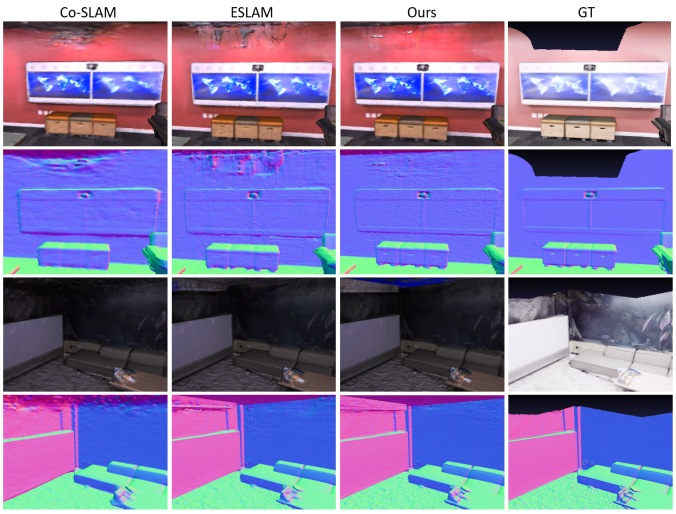

Fig. 6: The reconstruction results of unobserved regions. Benefit from regularization of structural consistency, our method achieves the most reasonable predictions.

Table 3: Runtime and computation analysis results on Replica office0 [30]. The best results are in bold.

	Tracking(s)↓	Mapping(s)↓	#Param↓	FLOPs[$\times 10^3$]↓
NICE-SLAM [43]	1.02	2.29	12.18 M	104.16
Point-SLAM [11]	0.85	9.85	27.23 M	–
ESLAM [11]	**0.14**	**0.52**	6.79 M	53.12
Ours	0.21	0.98	**2.43** M	**34.85**

point. Notably, because the neural point cloud in Point-SLAM [27] dynamically grows during running, its FLOPs cannot be calculated using a fixed value.

Scene Representation and Optimization Policy Tab. 4 (top) presents the evaluation of different scene representation policy on room0 of Replica [30] and scene0207 of ScanNet [5]. Our comprehensive model demonstrates higher reconstruction accuracy and more precise location compared to only using 2D multi-scale feature planes (a) or single 3D hash (b). We also examined the performance without TSDF features in the geometric prediction phase (c) like ESLAM [11].

As shown in Tab. 4 (bottom), our various optimization choices and their interpretations are as follows. (d) We do not consider color rendering and ignore the L_c (Sec.3.1).

(e) We do not employ structural consistency constraints and disregard the L_{con} (Sec.3.2). (f) We do not employ Manhattan matching constraints and neglect the L_{coor} (Sec.3.2). (g) We do not exploit smooth loss for the hash grid and ignore the L_{smo} (Sec.3.3). (h) We evaluate our full model.

Table 4: The results of Replica room0 and ScanNet scene0207 with various optimization configurations. The best results are in bold. For further ablation analysis and qualitative results, refer to the supplementary.

	room0			sc.207
Scene Representation	Acc.↓	Comp.↓	RMSE↓	RMSE↓
a. No Hash Grid	2.23	1.79	0.65	6.03
b. No Feature Plane	2.72	2.19	1.02	4.73
c. No TSDF Frature	2.28	1.83	0.60	4.62
Optimization Choice				
d. No Color Loss	2.24	1.78	0.63	4.71
e. No Stru. Consist. Loss	2.46	1.81	0.63	4.75
f. No Man. Match. Loss	2.36	1.85	0.68	4.95
g. No Smooth Loss	2.38	1.80	0.61	5.85
h. Full Model	**2.23**	**1.78**	**0.58**	**4.61**

5 CONCLUSION

This article presents iS-MAP, an RGB-D SLAM system based on neural implicit mapping for structural environments. Our method utilizes hash grid and multi-scale feature plane hybrid encoding to achieve a hierarchical scene representation that considers both 2D and 3D details. We also maintain the structural consistency of planes and lines and consider matching associations based on the Manhattan hypothesis to better suit structured scenes. We conducted comprehensive experiments on synthetic and real datasets, demonstrating that our method outperforms existing approaches in both reconstruction and camera tracking while also being competitive in terms of runtime and memory usage. Moreover, our method predicts the most reasonable reconstruction results for the unobserved structural regions.

iS-MAP still has shortcomings. The pre-processing plane-line segmentation of images consumes extra running time and reduces overall speed. Enhancing the efficiency of segmentation and accelerating processing is a future direction. Additionally, extending loop closure and addressing dynamic targets are also intriguing questions we hope to address in future work.

Acknowledgement. This work was supported by Foundation of Science and Technology Innovation Leading Talent Project of Special Support Plan for High-level Talents of Zhejiang Province (Grant No. 2022R52053).

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5855–5864 (2021) 3

2. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam—learning a compact, optimisable representation for dense visual slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2560–2568 (2018) 3
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021) 1
4. Chen, Z., Wang, C., Guo, Y.C., Zhang, S.H.: Structnerf: Neural radiance fields for indoor scenes with structural hints. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 3
5. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017) 9, 11, 12, 13
6. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Transactions on Graphics (ToG) **36**(4), 1 (2017) 3
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981) 6
8. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5511–5520 (2022) 3
9. Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T., et al.: Local implicit grid representations for 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6001–6010 (2020) 3
10. Jiang, H., Ding, L., Hu, J., Huang, R.: Plnet: Plane and line priors for unsupervised indoor depth estimation. In: 2021 International Conference on 3D Vision (3DV). pp. 741–750. IEEE (2021) 3, 6
11. Johari, M.M., Carta, C., Fleuret, F.: Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17408–17419 (2023) 2, 4, 5, 8, 9, 10, 11, 13
12. Kim, P., Coltin, B., Kim, H.J.: Linear rgb-d slam for planar environments. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 333–348 (2018) 3
13. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. pp. 225–234. IEEE (2007) 1
14. Li, M., He, J., Wang, Y., Wang, H.: End-to-end rgb-d slam with multi-mlps dense neural implicit representations. IEEE Robotics and Automation Letters **8**(11), 7138–7145 (2023) 2, 4
15. Li, R., Wang, S., Gu, D.: Deepslam: A robust monocular slam system with unsupervised deep learning. IEEE Transactions on Industrial Electronics **68**(4), 3577–3587 (2020) 3
16. Li, Y., Brasch, N., Wang, Y., Navab, N., Tombari, F.: Structure-slam: Low-drift monocular slam in indoor environments. IEEE Robotics and Automation Letters **5**(4), 6583–6590 (2020) 3

17. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021) 3
18. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1, 3
19. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022) 4
20. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015) 1
21. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics **33**(5), 1255–1262 (2017) 1
22. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: IEEE international symposium on mixed and augmented reality. pp. 127–136 (2011) 1, 3
23. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: International Conference on Computer Vision. pp. 2320–2327. IEEE (2011) 1, 3
24. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J.J., Kemelmacher-Shlizerman, I.: Stylesdf: High-resolution 3d-consistent image and geometry generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13503–13513 (2022) 5
25. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 523–540. Springer (2020) 3
26. Rosinol, A., Leonard, J.J., Carlone, L.: Nerf-slam: Real-time dense monocular slam with neural radiance fields. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3437–3444 (2023) 2
27. Sandström, E., Li, Y., Van Gool, L., Oswald, M.R.: Point-slam: Dense neural point cloud-based slam. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18433–18444 (2023) 2, 3, 4, 9, 11, 13
28. Schops, T., Sattler, T., Pollefeys, M.: Bad slam: Bundle adjusted direct rgb-d slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 134–144 (2019) 1, 3
29. Shu, F., Wang, J., Pagani, A., Stricker, D.: Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2105–2112 (2023) 3
30. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., et al.: The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019) 9, 11, 12, 13
31. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: imap: Implicit mapping and positioning in real-time. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6229–6238 (2021) 2, 4, 9, 10, 11
32. Teed, Z., Deng, J.: Deepv2d: Video to depth with differentiable structure from motion. arXiv preprint arXiv:1812.04605 (2018) 3

33. Teed, Z., Deng, J.: Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. Advances in neural information processing systems **34**, 16558–16569 (2021) 3
34. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. IEEE transactions on pattern analysis and machine intelligence **32**(4), 722–732 (2008) 6
35. Wang, B., Chen, C., Cui, Z., Qin, J., Lu, C.X., Yu, Z., Zhao, P., Dong, Z., Zhu, F., Trigoni, N., et al.: P2-net: Joint description and detection of local features for pixel and point matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16004–16013 (2021) 3
36. Wang, H., Cao, Y., Wei, X., Shou, Y., Shen, L., Xu, Z., Ren, K.: Structerf-slam: Neural implicit representation slam for structural environments. Computers & Graphics p. 103893 (2024) 2, 4, 6, 9, 10, 11
37. Wang, H., Wang, J., Agapito, L.: 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. pp. 13293–13302 (2023) 2, 4, 9, 10, 11
38. Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., Davison, A.: Elasticfusion: Dense slam without a pose graph. Robotics: Science and Systems (2015) 1, 3
39. Yang, X., Li, H., Zhai, H., Ming, Y., Liu, Y., Zhang, G.: Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 499–507 (2022) 2, 4, 9, 10, 11
40. Zhang, Y., Chen, G., Cui, S.: Efficient large-scale scene representation with a hybrid of high-resolution grid and plane features. arXiv preprint arXiv:2303.03003 (2023) 3
41. Zhou, H., Zou, D., Pei, L., Ying, R., Liu, P., Yu, W.: Structslam: Visual slam with building structure lines. IEEE Transactions on Vehicular Technology **64**(4), 1364–1375 (2015) 3
42. Zhu, S., Wang, G., Blum, H., Liu, J., Song, L., Pollefeys, M., Wang, H.: Sni-slam: Semantic neural implicit slam. arXiv preprint arXiv:2311.11016 (2023) 2, 4
43. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12786–12796 (2022) 2, 4, 9, 10, 11, 13