Accepted in IEEE International Conference on Robotics and Automation (ICRA) 2024

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

End-to-End RGB-D SLAM with Multi-MLPs Dense Neural Implicit Representations

Mingrui Li, Jiaming He, Yangyang Wang, Hongyu Wang

***Abstract*— An accurate and generalizable dense 3D reconstruction system has attracted much attention. However, existing 3D dense reconstruction systems are constrained by pre-training, and there is a need for enhanced reconstruction of texture and shape details. We propose an end-to-end 3D reconstruction system which achieves fine scene reconstruction without prior information by utilizing a neural implicit encoding. Our proposed system successfully achieves the goal through improved multi-MLP decoders (*MLM*) and an effective keyframe selection strategy. Experiments conducted on the commonly used Replica and TUM RGB-D datasets demonstrate that our approach can compete with widely adopted NeRF-based SLAM methods in terms of 3D reconstruction accuracy. Moreover, our approach shows a 40.8%(except Completion Ratio) improvement in accuracy compared to NICE-SLAM [14] and does not use prior information.**

I. Introduction

Realizing a dense simultaneous localization and mapping system (SLAM) based on RGB-D is a critical challenge in the fields of computer vision and robotics. This system exhibits broad applicability in domains such as autonomous driving, virtual reality, and robotics. Traditional SLAM methods are categorized into sparse SLAM [1], [2], dense SLAM [3], [4] and hybrid systems [5]. These SLAM methods [6], [8]–[10], [29] are plagued by limitations in sensor accuracy and require significant computational resources for feature-based matching and optimization algorithms, which limit their performance. A significant drawback is their inability to generate a dependable geometric estimation of the unobserved region [11].

In recent times, SLAM approaches utilize neural radiation fields and the iMAP [12], NeRF-SLAM [13], and NICE-SLAM [14] systems, have exhibited remarkable proficiency in reconstructing 3D surfaces and large scenes [15], [16]. However, neural implicit SLAM systems, such as NICE-SLAM requires pre-training on synthetic indoor scene datasets [17], limiting their generalization ability. Current neural implicit SLAM systems without pretraining, such as iMAP, exhibit limited reconstruction accuracy. Since a single MLP's global parameter optimization is frequently used, the system will suffer from forgetting.

Our approach utilizes multiple MLPs to extract detailed features. We achieve dense mapping by employing neural implicit reconstruction of color and occupancy. Our method avoids the limitation of a single MLP and propose a multi-MLP neural implicit encoding structure to extract detailed features, while achieving local updating and optimization. This improves the tracking and mapping strategies and leads to a higher quality reconstruction process. Compared to the baseline NICE-SLAM, we achieve improvements in reconstruction accuracy and quality without pre-training and can compete with the state-of-the-art RGB-D SLAM methods.

In summary, we put forward three significant contributions: (i) We propose an end-to-end dense SLAM system capable of achieving 3D reconstruction of virtual and real scenes without any prior infomation, while ensuring sufficient detail and scalability, (ii) To enhance the feature detail extraction capability of the system. We propose a multi-MLP neural implicit coding structure which can extract more detailed features, enabling a more comprehensive reconstruction process. (iii) To enhance the tracking stability of the system,we propose a keyframe selection strategy. The strategy based on sliding window filtering and can adaptively switch the keyframes involved in local optimization in the keyframe list.Extensive evaluations on the Replica dataset [20] and the TUM RGB-D dataset [21] demonstrate that our method competes well with similar approaches in terms of reconstruction accuracy, PSNR, and Depth L1. Compared to NICE-SLAM [14], we achieved at least a 40.8%(except Completion Ratio) improvement.

II. Related Work

Traditional SLAM methods [22]–[26] have signifcant achievements in dynamic tracking, sparse mapping, semantic analysis, and other aspects. Vision-based methods are more convenient and cost-effective compared to lidar-based methods. Dense SLAM systems, as opposed to sparse visual representations, allow for superior quality scene reconstructions.They also offer better occlusion reasoning, collision detection, interpretation of scene content and perception. Early KinectFusion [10] implemented point cloud reconstruction by establishing a voxel grid that could be updated gradually. Although the explicit method can achieve fine reconstruction results at the pixel level. Compared with neural implicit methods, the large amount of memory consumed by the saved reconstruction results is always a challenge. [27]. Researchers are therefore more inclined to use indirect methods to reduce the resource requirements.

In recent years, CodeSLAM [28] proposed an autoencoder that can split the high-level features of grayscale images through a pretrained network and realize hierarchical sampling to refine image features. Tandem [31] introduced a neural network into the dense reconstruction process,

All authors are with the Dalian University of Technology, China. Yangyang Wang is additionally with the Dalian Maritime University, China.

This work was supported in part by the National Natural Science Foundation of China under Grants 61671103 and is part by the Science and Technology Innovation Funds of Dalian under Grants 2022JJ11CG002. (Corresponding author: Hongyu Wang.)

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

which can use the pre-trained MVSNet [32] network for depth estimation and reduce drift in large scenes through frame to model tracking. DROID-SLAM [33] used a dense optical flow architecture to reduce reprojection errors while improving tracking robustness. In the above dense SLAM methods,point cloud or voxel representation are often necessary and the traditional methods are limited in their ability to fill occluded areas of the scene.

Neural implicit representations have shown remarkable success in achieving 3D reconstruction, [34]–[37], especially in filling unknown regions and depth recovery. The core of the method is the continuous representation of the signal that leads to excellent performance in object reconstruction and scene completion. However, current work mostly focus on small scenes and objects and falls short in larger-scale scenes. Moreover, most current Nerf require prior information of camera pose and lack the ability to achieve mapping-tracking processes [39]–[43]. Combining the neural implicit representations methods with SLAM is considered a crucial method for verifying Nerf performance and future applications [36], [44]–[47], but current Nerf work is generally regarded as an offline method due to its long cycle time.

iMAP [12] is the first work that successfully implements real-time online SLAM, realizing the global update and joint optimization process through a single MLP [48]. However, iMAP [12] consumes considerable memory and computing resources since it stores and compresses the cache state map file and performs periodic local enhancements. BARF [30] is a similar work to view synthesis, and its bundled iterative process can obtain precise poses meeting the Nerf reconstruction requirements.

NICE-SLAM [14] adopts a hierarchical storage scene representation, and stores scene partitions through a pre-trained differentiable renderer. However, due to the frozen pre-training parameters and low-resolution storage structure, its generalization ability is limited in some scenarios and may lose some details. Moreover, the running speed of the system is limited by different scenarios. Our method does not require pre-training or pre-setting of reconstruction range parameters. But can also achieve the filling of unknown areas, improving the geometric integrity of the reconstruction, while providing reasonable photometric predictions.

III. Problem Statement

IV. Approach

We have developed a hierarchical 3D reconstruction system based on the neural implicit representation to enhance the system's perceptual ability and feature extraction capability for fine details in the scene. Specifically, our approach does not depend on pre-trained models. To improve feature extraction, we have created a multi-MLP differentiable rendering framework for layered scene representation. Furthermore, we have designed a keyframe selection strategy that can enhance the accuracy of tracking. Specifically, in Section A, we explain the workflow of our neural implicit rendering method as shown in Fig1. In Section B, we introduce the architecture and implementation of our multi-MLP hierarchical feature extractor. Finally, in Section C, we provide details of our active evaluation and optimization strategy.

A. Multiscale Hierarchical MLP Residual Structured Scene Representations

iMAP [12] is limited in performance when facing large-scale scenes due to the storage capacity limitation of a single MLP. At the same time, the simple feature concatenation as input to a multi-MLP as done by NICE-SLAM [14] is also difficult to fully solve the problem of extracting detailed features in the scene. Therefore, we propose a multi-MLP-based SLAM system in this paper. Our multi-MLP scene renderer, ***MLM*** (MLP with Large and Minor), includes multiple levels of decoders.

The construction leads to the total calculation amount has not significantly increased. In the experiment, we utilize four voxel grids with side lengths of 32, 24, 16 (The corresponding feature is represented as $\xi_c^1\ \xi_c^2\ \xi_c^3\ \xi_c^4$), and 8cm as the geometric division of the MLP interpreter.We utilize two parallel structures, referred to the large layers and the minor layers. Both the large layer and the minor layer are composed of a perception structure consisting of two MLPs.

Because we propose a relatively lightweight MLP structure, despite increasing the number of MLPs, the total number of layers has decreased. These decoders are divided into two groups: large and minor scales, corresponding to different levels of MLP interpreters. We use $\boldsymbol{MLM^{large1}}$, $\boldsymbol{MLM^{large2}}$, $\boldsymbol{MLM^{minor1}}$ and $\boldsymbol{MLM^{minor2}}$ to represent it. The two MLPs in the Large-MLPs are designed to provide residuals ($\Delta F_\zeta^{3\text{large}}$ and $\Delta F_\zeta^{4\text{large}}$) the Minor-MLPs with a feature grid side length of 16cm. These settings leads to facilitate the learning of richer high-frequency details. The Large-MLPs do not participate in detailed color rendering or occupancy value output. The MLP design is flexible, which allows our architecture to have certain scalability and structural flexibility. The occupancy offset of $\boldsymbol{MLM^{minor2}}$ layer can be obtained by this process:

$$\Delta F_\xi^{2\text{large}} = \boldsymbol{MLM^{minor2}}\left(\Delta F_\xi^{4\text{large}} + \Delta F_\xi^{3\text{large}}\right) \tag{1}$$

In the $\boldsymbol{MLM^{large1}}$ and $\boldsymbol{MLM^{large2}}$ layers, we apply two transformations to each residual block with $\alpha = 1, \beta = 0$ and use them to control the output proportion and bias correction. After learning the residual, we input it into $\boldsymbol{MLM^{minor1}}$ for feature extraction and perform another residual pass. The performed residual fusion is expressed as $\Delta F_\xi^{4\text{large}} + \Delta F_\xi^{3\text{large}}$. We input the residual of $\boldsymbol{MLM^{minor1}}$ into $\boldsymbol{MLM^{minor2}}$ to enhance the extraction of details, compensating for high-frequency features and avoiding the automatic filtering problem of MLP. This relies primarily on the reinforcement obtained from the stacked features of the fine-grained layers. Finally, we use $\boldsymbol{MLM^{minor2}}$ as the main reconstruction interpreter, outputting both occupancy values and colors.

$$\Delta O_\delta^{\text{minor2}} = \boldsymbol{MLM^{minor2}}\left(\Delta F_\xi^{4\text{large}}; \Delta F_\xi^{3\text{large}}\right) \tag{2}$$

$$\Delta O_\delta^{\text{minor1}} = \boldsymbol{MLM^{minor2}}\left(\Delta F_\xi^{2\text{minor}}\right) \tag{3}$$

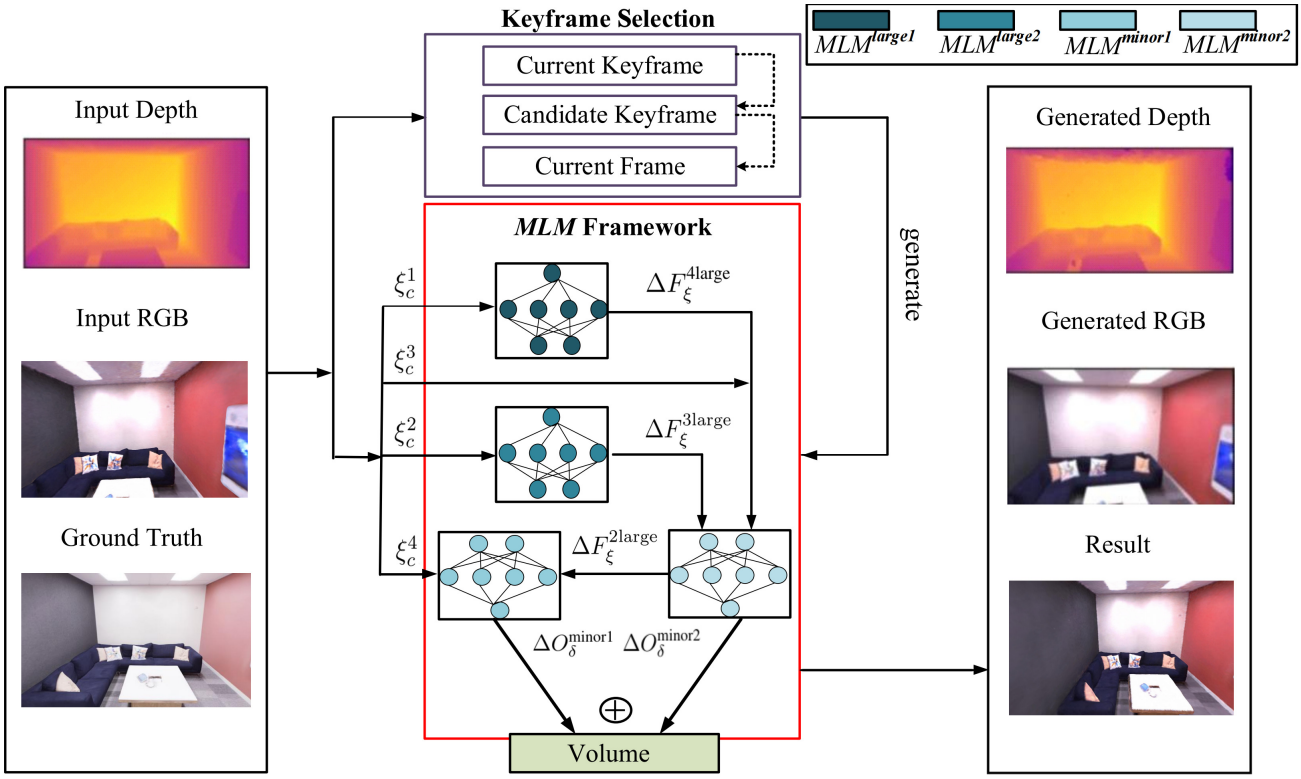

Fig. 1: The structure of the our system is presented in Figure 1, which depicts an end-to-end parallel tracking and mapping process. The input to the pipeline is a continuous RGB-D image stream, and the output is the reconstruction result and model. To extract features hierarchically, we use a feature encoder with 4 MLPs and sample the input image. The two MLP decoders in the large-MLP layer only perform residual transfer and do not participate in the voxel reconstruction process. The feature residuals extracted are input into the coarse decoder of the small-MLP layer, where the two decoders jointly participate in color and volume reconstruction and evaluate the reconstruction quality simultaneously. The keyframe selection strategy can be adaptively adjusted to rationalize the bundle adjustment process and improve tracking accuracy. The MLP decoders of different scales are represented by a model from deep to shallow.

The final representation of the occupancy value for a point is:

$$O_\delta = \Delta O_\delta^{\text{minor1}} + \Delta O_\delta^{\text{minor2}} \tag{4}$$

Both $\Delta F_\xi^{4\text{large}}$ and $\Delta F_\xi^{3\text{large}}$ are fed as residual inputs, and simultaneously participate in optimizing the feature grid. We use multi-level MLP interpreters to directly obtain occupancy values, resulting in a more fexible and generalizable network structure. It is a process of deepening feature learning multiple times, and through experiments, it has been proven that it is feasible to approach the performance. Specifically, when voxel content O_δ^0 cannot be filled in the minor layer, we use a predicted value obtained directly from $\boldsymbol{MLM^{large1}}$ and $\boldsymbol{MLM^{large2}}$ of the Large-MLPs layer to fill it. Occupied values are then interpolated using bilinear interpolation to effectively fill holes in unknown regions. To expand the observation range in larger scenes, we can adjust the parameters of the Large-MLPs layer.

$$O_\delta^0 = \boldsymbol{MLM^{large1}}\left(O_\delta^{\text{large1}}\right) \tag{5}$$

B. Neural Implicit Representations

We use pixel sampling on the current frame and initialize the camera pose using a constant speed assumption, and update the frame pose through backpropagation. To enhance system speed we employ hierarchical volume sampling strategy [49].First, we sample N_s along each ray, and then additionally sample N_i points near the surface. Overall, we sample $N_t = N_s + N_i$ points.For all sampling points P_i on the given ray $\mathbf{k}$. we have $p_i = O + Z_i(\mathbf{k})\ \mathbf{i} \in \{1, \cdots, \mathrm{N}\}$, where O is the center coordinate of the camera and $Z_i(\mathbf{k})$ is the depth at the i-th sampling point on the ray $\mathbf{k}$ or every point p_i , we can calculate their large-level occupancy probability O_δ^0 minor-level occupancy probability O_δ we model the ray termination probability at point p_i as $W_m^{\mathrm{k}} = O_\delta^0 \prod_{j=1}^{i-1}\left(1 - O_\delta^0\right)$ for large level, and $W_i^k = O_\delta \prod_{j=1}^{i-1}(1 - O_\delta)$ for minor level. For each ray, the depth at both coarse and fine level:

$$C_m(k) = \sum_{n=1}^{N} W_m^k Z_i(k) \tag{6}$$

$$C_1(\mathrm{k}) = \sum_{n=1}^{N} W_j^{\mathrm{k}} Z_{\mathrm{i}}(k) \tag{7}$$

The volume density is used to render the color and depth of each ray. Unlike NICE-SLAM [14], we have abandoned the fixed pretrained decoder that needs to be pretrained in large scenes. This allows us to avoid learning specific resolution priors, which leads to better generalization of our system. NICE-SLAM [14] runs at significantly different speeds in scenes of different scales, whereas our system has better generalization performance, which can significantly alleviate this problem. The geometric and photometric losses are expressed as follows, to maintain consistency:

$$\mathcal{L}_g = \frac{1}{N_{\mathrm{t}}} \sum_{m=1}^{N_{\mathrm{t}}} \left| C_{\mathrm{m}}(k) - \hat{C}_{\mathrm{m}}(\mathrm{k}) \right| \tag{8}$$

$$\mathcal{L}_{\mathrm{l}} = \frac{1}{N_{\mathrm{t}}} \sum_{m=1}^{N_{\mathrm{t}}} \left| C_{\mathrm{m}}(k) - \hat{C}_{\mathrm{m}}(\mathrm{k}) \right| \tag{9}$$

where $\mathcal{L}_g$ and $\mathcal{L}_l$ denote geometric and photometric losses. Both geometric and photometric losses at the large and minor levels are L_1 losses between the predicted and observed values. To obtain the depth of each pixel, we sample the position and volume density along the ray of the pixel and perform alpha synthesis on the density obtained. To minimize the depth and color loss of the pass, we optimize the process by minimizing the photometric and geometric errors. In the optimization of the feature grid, we use Eq. (1) to optimize the large and minor levels sequentially, and then perform BA process including: the feature grid corresponding to the minor level, as well as the camera extrinsics and the filtered keyframes. For the feature grid at the large level, we optimize the weighted sum of both losses using an adjustable λ_p to adjust the weight of photometric error in the minor level.

$$\mathcal{L}_{lm} = \min \left\| \mathcal{L}_1 + \mathcal{L}_z + \lambda_p \mathcal{L}_l \right\|_2 \tag{10}$$

We optimize only the Minor-MLPs layer, and perform local bundle adjustment as a final step. The optimization process is localized to minimize the reprojection error. Because the texture and color details corresponding to the feature grid at the large level are too coarse, we try to minimize its impact while speeding up the optimization process.

C. Tacking and Mapping methods

Tracking: Our tracking approach is designed to operate in parallel, with inter-thread data exchange only occurring during keyframe generation. To facilitate differentiable rendering, we employ feature embedding in lieu of global coordinates and conduct iterative optimizations for both depth and geometry [51]. Pose updates are measured within the tangent space of SE3. To mitigate the issue of tracking drift, we randomly sample and extract a subset of pixels from the keyframe to form a pixel set of size N_{t}. During tracking, we remove pixels whose color loss exceeds eight times the mean value, as these pixels tend to be unstable.

Mapping: In terms of selecting keyframes, NICE-SLAM simply extracts a keyframe every 50 frames and performs BA optimization after 4 keyframes. Compared to NICE-SLAM, we propose a more refined strategy.In order to make our strategy more reasonable during the BA (bundle adjustment) process and avoid the low quality of selected keyframes affecting the tracking and mapping process, we choose one frame as a keyframe every 20 frames and perform BA optimization every 4 frames. At the same time, we take the last two frames as candidate keyframes and observe that when the pixel loss is too high or the illumination changes drastically, the current frame is often in a high dynamic state, which may lead to tracking failure and low reconstruction accuracy. Therefore, we divide the current frame into a 20×20 grid and calculate the median depth loss ϕ of each grid, and then calculate the total depth loss median ϕ_t. We remove all pixels in the current frame greater than 8 times the median pixel loss of all pixels, and if we detect that 5% of the pixels in the current frame are greater than 8 times the median depth loss or the total number of grids with luminosity changes greater than the median exceeds 5, we consider the current frame unreliable and remove it from the keyframe list, adding the candidate keyframe to the list. However, if the candidate keyframe still does not meet the threshold requirements after switching, we directly select the current frame for generation and tracking, and perform local BA. This is particularly important in real-time scenarios, such as challenging dynamic scenes. In this case, we turn off the re-projection process and directly use the current frame to accelerate the mapping process. This method can be a basis for future semantic SLAM, but it is not the focus of this paper. The experiments demonstrate that we can achieve complete tracking process. This is especially crucial when there are real-time requirements, such as challenging dynamic scenes. In such cases, we turn off the regeneration process and directly use the current frame to speed up the tracking process.

V. Experiments

We conducted an evaluation of both competing methods and our proposed approach. We conducted experiments on both real-world and synthetic datasets and measured relevant evaluation metrics to assess the performance of each method. Furthermore, we conducted ablation experiments to analyze the contribution of individual components in our proposed architecture and to quantify the improvement achieved. The results of our evaluation provide valuable insights into the effectiveness of our approach and its potential for practical applications.

A. Experimental Setting

To perform the reconstruction task, we use PyTorch and CUDA, and performed the computation on an i7-12700K CPU and a 3090ti GPU equipped with 24GB of video memory. Specifically, we adopted a weighted photometric loss with a weighting parameter λ=0.2 on the TUM RGB-D dataset. For each image, we sampled N = 1000 and N_{t} = 200 pixel values, respectively.

Datasets: We utilized three Datasets for our evaluation: the Replica dataset, ScanNet dataset [17] and TUM RGB-D dataset [21]. The Replica dataset is a synthetic 3D scene dataset that contains eight sequences, each with 2000 RGB and depth images. This Dataset includes five office environments and three apartment environments. The ScanNet dateset and TUM RGB-D dataset are collected from multiple sensors, which consists of three sequences that can effectively demonstrate the system's reconstruction and tracking capabilities in challenging environments.

Baseline and Metrics: We followed the same MLP settings as NICE-SLAM and modified the code to remove the pre-training part as control. As we observed that the non-pretraining version of NICE-SLAM has a significant random error, we used the best data from 10 experiments as its performance index, while the data of NICE-SLAM refer to the data provided by the authors in their paper. To comprehensively evaluate the reconstruction accuracy, we selected accuracy (cm), completion(cm), and completion

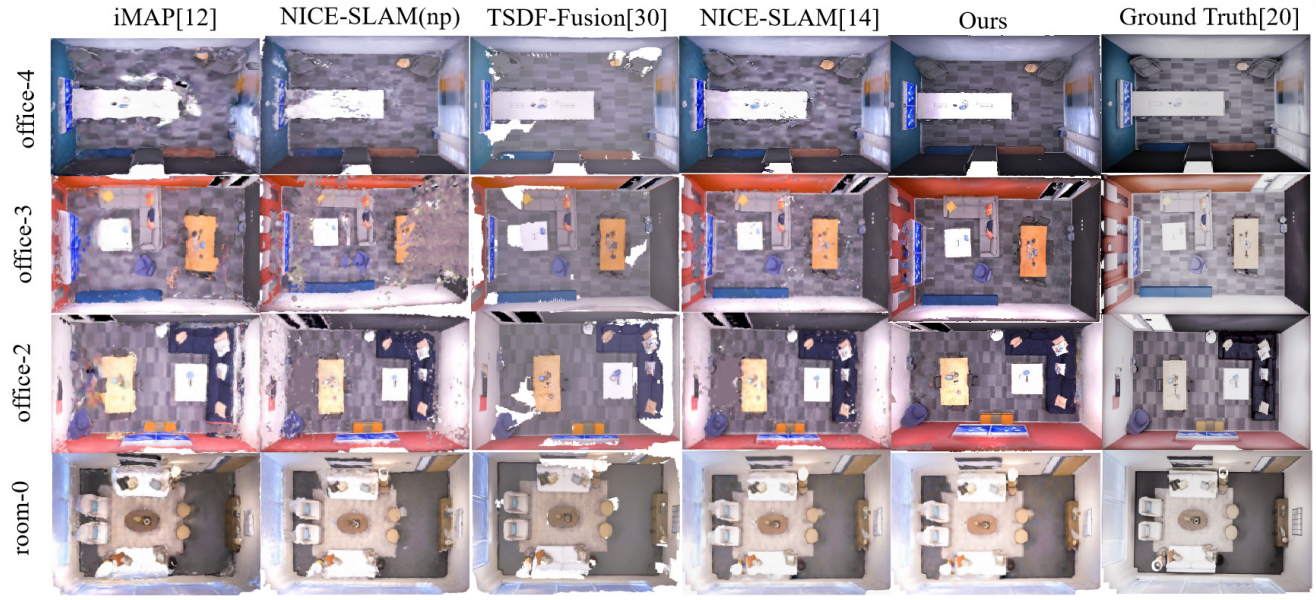

Fig. 2: Reconstruction Results of 8 scenes in the Replica dataset.The order of the methods in the image, from left to right, is as follows: iMAP NICE-SLAM (np), TSDF-Fusion NICE-SLAM, our proposed approach, and ground truth.The TSDF reconstruction results were obtained using Tandem. For IMAP we used the code reproduced in NICE-SLAM. According to the reconstruction results of Fig2 that our reconstruction results have higher resolution.(We use NICE-SLAM(np) to represent the non-pretrained version of NICE-SLAM)

TABLE I: Reconstruction Results of 8 Scenes in the Replica dataset. Compared with IMAP NICE-SLAM and NICE-SLAM (np). We evaluate the reconstruction quality and assess the reconstructed geometry and image quality using Geometric (Depth L1) and Photometric (PSNR) metrics on the Raplica dataset.The best results were highlighted in bold.

		office0	office1	office2	office3	office4	room0	room1	room2	AVG
NICE-SLAM	Acc.[cm]↓	5.56	3.35	4.71	**3.84**	**3.35**	3.53	3.60	**3.03**	3.87
	Comp.[cm]↓	4.55	4.03	3.94	3.99	3.87	3.40	3.62	3.27	3.87
	Comp. Ratio[<5cm%]↑	89.30	88.79	88.97	**87.18**	**91.17**	91.92	91.36	**90.79**	89.93
	Depth L1[cm]↓	2.62	2.91	8.14	5.47	**2.25**	2.53	3.43	**2.96**	3.79
	PSNR[dB]↑	25.78	25.30	18.50	22.82	25.26	23.83	**22.61**	21.97	23.26
iMAP	Acc.[cm]↓	5.87	3.71	4.81	4.27	4.83	3.58	3.69	4.68	6.95
	Comp.[cm]↓	6.11	5.26	5.65	5.45	6.59	5.06	4.87	5.51	5.33
	Comp. Ratio[<5cm%]↑	77.71	79.64	77.22	77.34	77.63	83.91	83.45	75.53	79.06
	Depth L1[cm]↓	6.43	7.41	14.23	8.68	6.80	5.70	4.93	6.94	7.64
	PSNR[dB]↑	7.39	11.89	8.02	5.62	5.98	5.66	5.31	5.64	6.95
NICE-SLAM(np)	Acc.[cm]↓	5.01	**3.10**	8.46	4.73	16.1	4.22	12.5	3.32	7.15
	Comp.[cm]↓	5.03	4.58	5.85	3.55	10.8	4.26	7.44	4.27	5.72
	Comp. Ratio[<5cm%]↑	91.99	**89.53**	63.19	84.69	48.72	90.30	55.32	89.50	77.72
	Depth L1[cm]↓	1.83	3.01	13.89	6.47	19.20	4.23	10.96	2.81	7.61
	PSNR[dB]↑	22.09	21.63	20.24	21.21	10.83	18.51	17.91	16.93	18.66
Ours	Acc.[cm]↓	**2.48**	4.05	4.61	4.04	3.78	**3.27**	3.63	3.26	**3.64**
	Comp.[cm]↓	**2.13**	**3.41**	**2.99**	**3.39**	**2.93**	**2.66**	**2.15**	**2.66**	**2.80**
	Comp. Ratio[<5cm%]↑	**93.54**	85.62	**89.00**	85.63	88.53	**93.85**	**96.52**	90.58	**90.43**
	Depth L1[cm]↓	**1.70**	**2.36**	**2.09**	**2.10**	2.71	**2.14**	**1.77**	3.53	**2.3**
	PSNR[dB]↑	**29.61**	**26.15**	**26.35**	**23.54**	**25.95**	**28.02**	20.36	**27.139**	**26.27**

ratio(<5cm %) as indicators. We also used PSNR (dB) and Depth L1 (cm) to evaluate the comprehensive indicators of the new view.

B. Results on Replica

The reconstruction results of Replica dataset are presented in Table 1 and Fig2. We present the results of our experiment by highlighting the best and second-best performance values in bold and underlined, respectively. The 2D metrics were evaluated based on the PSNR and Depth L1 values, which were computed as the average of 8 scenes. Additionally, we have provided reconstruction results on four different scenes. Our method outperforms TSDF-Fusion resulting in clearer and more detailed geometry. This is mainly due to the large amount of feature information obtained and the holes filled in the reconstruction process. Our approach exhibits a more significant advantage over NICE-SLAM when pre-trained geometric priors are not available, achieves improvement of each performance values at least 40.8%(except Completion Ratio).

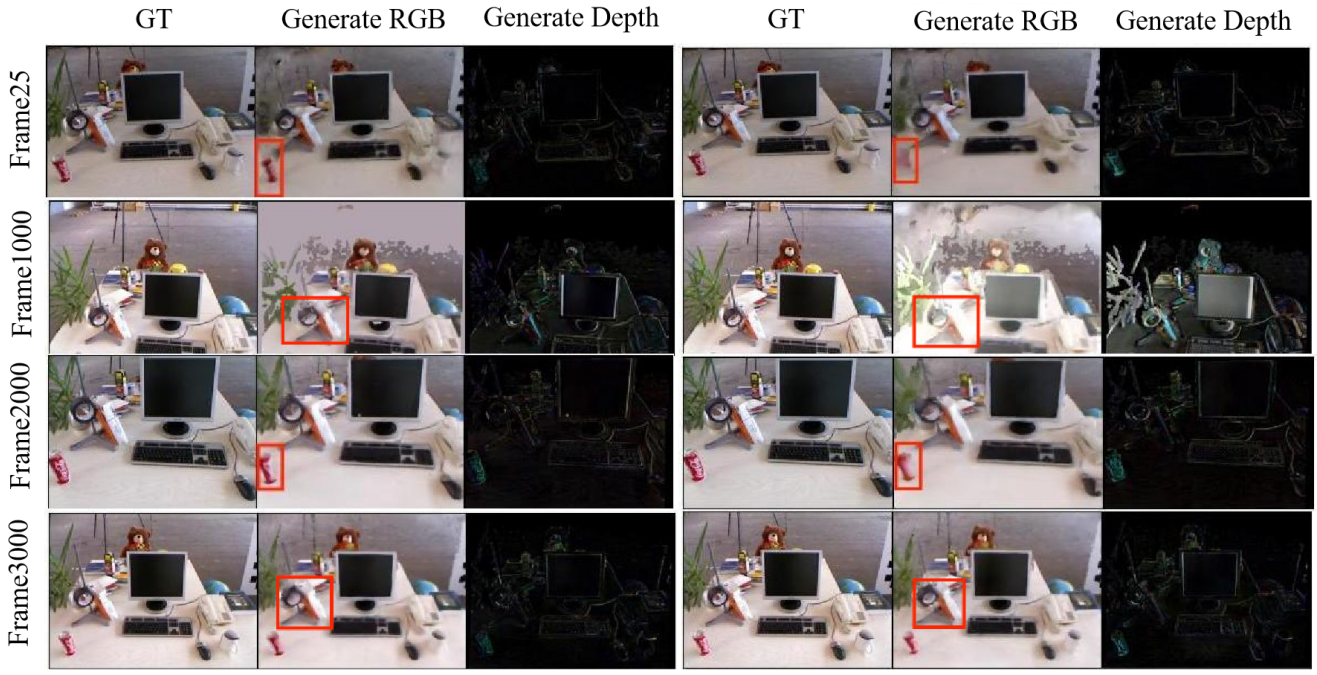

Fig. 3: TUM RGB-D dataset 25-3000 Frame Generation results.We compared the results generated by our method(left) and NICE-SLAM(right) on TUM RGB-D [21] dataset. The results from frame 25 to frame 3000 are shown from top to bottom. Details for comparison are highlighted with red boxes in the figures. This comparison was conducted following academic standards.

C. Results on TUM RGB-D

The results of ATE in TUM RGB-D datasets is shown in Table2. The TUM RGB-D dataset [21] comprises real-world scenes representing a challenging benchmark for Nerf-based SLAM approaches. We evaluate the performance of NICE-SLAM, DI-fusion [19], BAD-SLAM [54], iMAP [12], and NICE-SLAM (np) on the TUM RGB-D dataset. Our results indicate that, while our approach falls short of BAD-SLAM in terms of tracking accuracy, it outperforms NICE-SLAM, iMAP, and DI-fusion.Figure 3 shows the generation results on the TUM RGB-D dataset, including frame 25, frame 1000, frame 2000, and frame 3000. We adopted the same minor layer parameter settings as NICE-SLAM on the TUM RGB-D dataset [21] to ensure fairness. Additionally, we presented Ground Truth, Generate RGB and Generate Depth images. The contrast effect of local detail reconstruction in the red box, our method from the depth in terms of color reconstruction results, we are richer in details than NICE-SLAM. The reconstruction effect of the 1000th frame also demonstrates that our method is more robust and can reconstruct geometric details more accurately.

D. Results on Scannet

The results of ATE in Scannet datasets is shown in Table3 and the results of run-time and memory comparison in Table4. We conduct the process speed and ATE tracking experiments on the Scannet dataset, and compared our method with NICE-SLAM [14], iMAP [12], and the latest ESLAM [7]. The speed testing experiment was based on the results of Scene0000. The results show that our method has advantages in terms of speed, and the final memory consumption is also relatively low due to the use of a relatively small network structure. We provide a comparison between the tracking results on Scene0000 and NICE-SLAM in Fig4.In terms of tracking, we are competitive with NICE-SLAM and ESLAM, and we achieve better results compared to the case where no keyframe selection strategy is used. Our method is capable of effectively reducing drift while also decreasing the parameter count. By flexibly adjusting the structure of the MLP, we can have a wider range of feature concatenation methods and keyframe selection thresholds tailored to different datasets. However, these aspects are not the focus of this paper.

TABLE II: Run-time and memory comparison on ScanNet [17] with respective settings. All methods are benchmarked on scene0000 of ScanNet.The best were highlighted in bold.

Scene ID	Sc. 0000	Sc. 0059	Sc. 0106	Sc. 0169	Sc. 0181	AVG
NICE-SLAM	12.4	15.2	7.9	10.9	13.1	11.9
ESLAM	7.3	**8.5**	7.5	6.5	9.0	7.8
iMAP	40.2	15.2	13.3	33.8	21.6	24.0
Ours(ns)	16.2	13.4	8.4	10.2	15.3	12.7
Ours	**6.9**	9.1	**7.4**	**3.1**	**8.6**	**7.0**

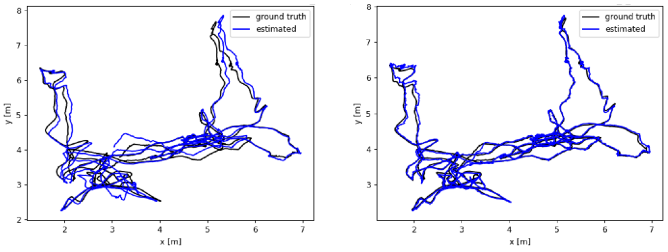

Fig. 4: Tracking Results on ScanNet [17] scene0000 sequence of ATE [cm] (↓), a comparison was made between NICE-SLAM and our method. Our method effectively reduces drift and enhances tracking accuracy.

TABLE III: Run-time and memory comparison on ScanNet [17] with respective settings. All methods are benchmarked on scene0000 of ScanNet.The best were highlighted in bold.

	Track(ms)↓	Map.(ms)↓	FPS↑	#param.↓
NICE-SLAM [14]	12.3×50	125.3×60	0.68	22.04
iMAP [12]	30.4×50	44.9×300	0.37	**0.22**
ESLAM [7]	N/A	N/A	1.82	17.63
ours	**7.9×50**	**30.2×10**	**4.4**	4.3

VI. Conclusion

We propose an end-to-end dense 3D reconstruction system utilizing neural implicit representation SLAM method. Our method employs MLM decoder to facilitate the mapping and tracking process, resulting in a hierarchical scene representation. Compared to pre-trained NERF combined with SLAM method, our method achieves excellent reconstruction accuracy without pre-training and preserves high-frequency details.Keyframe selection strategy adjustment can more effectively eliminate unreasonable keyframes, enhance tracking process stability, and achieve adaptive adjustments. Experimental results demonstrate that our method has advantages in both running speed and positioning accuracy and is competitive with recent approaches, while excelling in object-level details. Our system architecture has high adaptability, and we recommend further research focusing on improving voxel accuracy, varying MLP quantity, and input structure to achieve better results.

References

[1] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 7322–7328.

[2] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. M. Rehg, and J. Kautz, "Learning Rigidity in Dynamic Scenes with a Moving Camera for 3D Motion Field Estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 468–484.

[3] 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,pp. 1-2.

[4] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," *in Proc. Workshop RGB-D, Adv. Reason. Depth Cameras*, 2012, article 4..

[5] Handa, Ankur, Thomas Whelan, John McDonald, and Andrew J Davison. "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM." In *2014 IEEE international conference on Robotics and automation (ICRA)*, pp. 1524-1531. IEEE, 2014.

[6] R. Mur-Artal and J. D. Tard'os, "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, IEEE.

[7] 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*, 2023, pp. 17408-17419.

[8] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J. Davison, "DeepFactors: Real-Time Probabilistic Dense Monocular SLAM," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 721–728, 2020.

[9] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1168-1174.

[10] 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*, 2011, pp. 127-136.

[11] M. Strecke and J. Stuckler, "Em-fusion: Dynamic object-level slam with probabilistic data association," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5865-5874.

[12] 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.

[13] 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.

[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. 12786–12796.

[15] Y. Ming, W. Ye, and A. Calway, "iDF-SLAM: End-to-End RGB-D SLAM with Neural Implicit Mapping and Deep Feature Tracking," *arXiv preprint arXiv:2209.07919*, 2022.

[16] Hengyi Wang, Jingwen Wang, 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 (CVPR)*, 2023, pp. 13293-13302.

[17] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[18] K. Li, Y. Tang, V.A. Prisacariu, and P.H.S. Torr. "Bnv-fusion: dense 3D reconstruction using bi-level neural volume fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6166–6175.

[19] 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.

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

[21] J. Sturm, W. Burgard, and D. Cremers. "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*, 2012, pp. 1–7.

[22] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. "DTAM: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 2320–2327.

[23] 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.

[24] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang. "So-slam: Semantic object slam with scale proportional and symmetrical texture constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[25] R. Tian, Y. Zhang, Y. Feng, L. Yang, Z. Cao, S. Coleman, and D. Kerr. "Accurate and Robust Object SLAM with 3D Quadric Landmark Reconstruction in Outdoor Environment," in *IEEE Robotics and Automation Letters*, pp. 1534-1541, 2022.

[26] B. Bescos, C. Campos, J.D. Tard'os, and J. Neira. "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5191–5198, 2021.

[27] M. R"unz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4471–4478.

[28] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. "CodeSLAM—learning a compact, optimisable representation for dense visual SLAM." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2560-2568. 2018.

[29] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019, IEEE.

[30] 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.

[31] R. Craig and R. C. Beavis, "TANDEM: matching proteins with tandem mass spectra," *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004.

[32] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent MVSNet for high-resolution multi-view stereo depth inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5525–5534.

[33] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and RGB-D cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16558-16569, 2021.

[34] 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.

[35] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "Pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578-4587.

[36] Y.-C. Guo, D. Kang, L. Bao, Y. He, and S.-H. Zhang, "Nerfren: Neural radiance fields with reflections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18409-18418.

[37] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann. "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438-5448.

[38] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[39] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "Nerf in the dark: High dynamic range view synthesis from noisy raw images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16190–16199.

[40] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10318–10327.

[41] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[42] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "D^2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video," *arXiv preprint arXiv:2205.15838*, 2022.

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

[44] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "Humannerf: Free-viewpoint rendering of moving people from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16210–16220.

[45] 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. 12922–12931.

[46] D. Rebain, M. Matthews, K. M. Yi, D. Lagun, and A. Tagliasacchi. Lolnerf: Learn from one look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1558–1567, 2022.

[47] N. Pearl, T. Treibitz, and S. Korman. Nan: Noise-aware nerfs for burst-denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12672–12681, 2022.

[48] T. Pire, T. Fischer, G. Castro, P. De Crist'oforis, J. Civera, and J. J. Berlles. S-PTAM: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93:27–42, 2017.

[49] S. Zakharov, W. Kehl, A. Bhargava, and A. Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12224–12233, 2020.

[50] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. NeRF-SR: High Quality Neural Radiance Fields using Supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages=6445–6454, 2022.

[51] Jeffrey Ichnowski, Yahav Avigal, Justin Kerr, and Ken Goldberg. Dexnerf: Using a neural radiance field to grasp transparent objects. *arXiv preprint arXiv:2110.14217*, 2021.

[52] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022.

[53] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. NeRF-RPN: A general framework for object detection in NeRFs. *arXiv preprint arXiv:2211.11646*, 2022.

[54] Thomas Schops, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct RGB-D slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages=134–144, 2019.

[55] Wei-Cheng Tseng, Hung-Ju Liao, Yen-Chen Lin, and Min Sun. Clanerf: Category-level articulated neural radiance field. In *2022 International Conference on Robotics and Automation (ICRA)*, pages=8454–8460, 2022. IEEE.