StaticNeRF: Neural Implicit Static Mapping and Localization in Dynamic Environments

Juhui Lee[1], Seungjun Ma[1], Geonmo Yang[1] and Younggun Cho[1†]

***Abstract*— Recently, neural implicit representations have been widely introduced for robot mapping to achieve high-resolution maps. Previous approaches perform well in stable, and static environments but encounter difficulties when faced with the challenges posed by moving objects. In this paper, we propose a entire pipeline for neural implicit mapping and robust filter-based localization in dynamic environments. The entire scene can be decomposed into static and transient fields by implicitly learning geometric information, without the need for any external data. Moreover, this separation facilitates robust localization in dynamic environments by integrating a localization pipeline specifically tailored to the static field. Our approach is validated against standard and custom datasets, demonstrating that our implicit neural map has better performance than the other neural rendering methods and that our pipeline is effective in dynamic object removal and accurate in localization, marking a step forward for efficient navigation systems.**

I. Introduction

Visual localization is a critical technology across several domains, including robotics, augmented reality (AR), and autonomous vehicles. It enables precise positioning within an existing map, allowing mobile devices to interact seamlessly with the real world and provide valuable services to humans.

A significant development in this field is the creation of neural-based dense maps for localization which can generate photo-realistic scenes from novel viewpoints. NeRF [1], a representative of implicit neural rendering, can produce continuous information using only MLP network, in contrast to explicit representations that generate discrete one. Since the former one does not directly store 3D points, memory problem can be solved which is an essential consideration in robotics task, and still a limitation of explicit map representations. Moreover, the framework of NeRF [1] can be extended to incorporate additional functionalities such as semantic segmentation [2], and the disentanglement of object categories [3], making it a versatile tool for various applications.

One of the challenges in vision-based mapping systems is the requirement for data devoid of dynamic objects, as their presence can degrade the quality of the map. Since acquiring completely static map data is often impractical, there's a need to handle dynamic objects in the input images effectively. Traditional approaches have relied on supplementary modules for detecting dynamic objects, such as [4, 5], which identified moving objects and reconstructed the background scene using camera motion and static segmentation in RGB-D images.

Likewise, in traditional approaches, removing dynamic objects is crucial in NeRF [1] for improved outcomes. STaR [6] necessitated the use of multi-view video, enabling a geometrically convenient decomposition but impractical for the mapping process in robotics. DynamicNeRF [7] and NSFF [8] required external inputs like semantic segmentation or optical flow. NeRF-W [9] and NeuralDiff [10] segregated static and dynamic networks, training them with an aleatoric uncertainty loss. All these methods employed frequency encoding [1] leading to a larger network that is unsuitable for robotics application.

In this paper, we propose an implicit neural-based static map reconstruction method suitable for robotics. The main contributions of this work are the following.

1) We construct an implicit neural map optimized for robotics, focusing on minimizing training and inference time. This is achieved through the utilization of multiresolution hash encoding [11] based on an efficient sampling method.

2) Leveraging the capabilities of implicit representations, we introduce a mapping pipeline that effectively differentiates between static and dynamic elements without relying on additional external data.

3) We validate our method on both public and custom datasets, demonstrating effective dynamic object removal and improved localization accuracy through quantitative and qualitative evaluations.

II. Proposed Method

A. Problem Formulation

Our research introduces an implicit neural-based mapping and localization framework optimized for dynamic environments. We conceptualize the mapping problem as the process of learning a transformation from source domain X, which refers to train datasets obtained in a dynamic scene, to two target domains: a static domain S and a dynamic one D. Our mapping methodology can be represented as a function as follows: $G : X \rightarrow \{S, D\}$.

B. Network Structure

Our mapping algorithm is designed to utilize the concept of implicit, representing static and dynamic scene separately

[1]Juhui Lee, [1]Seungjun Ma, [1]Geonmo Yang and [1†]Younggun Cho are with the Electrical and Computer Engineering, Inha University, Incheon, South Korea [wngml6635, richard7714, ygm7422]@inha.edu, yg.cho@inha.ac.kr

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (RS-2023-00302589 and No.2022R1A4A3029480) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00448).

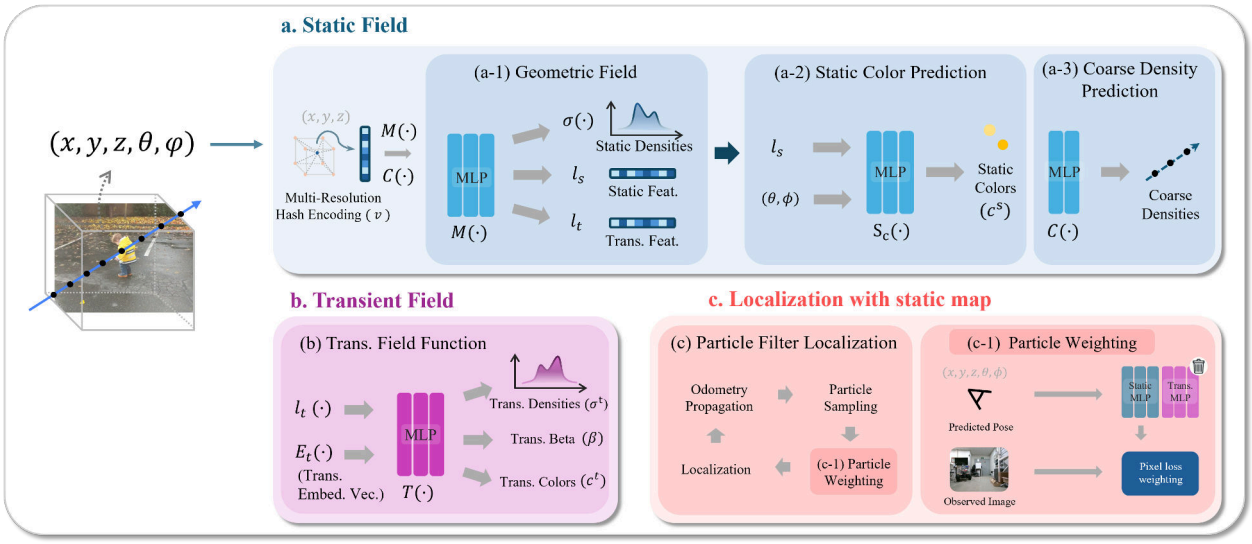
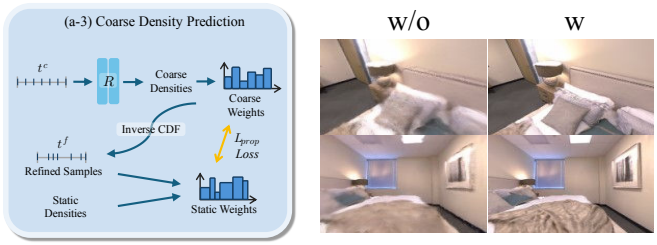


Fig. 1. Our advanced pipeline unfolds in two key phases. Initially, the process commences with mapping that leverages implicit neural representations to delineate both static and transient fields. Subsequently, localization tasks are precisely executed on the static map, ensuring robust to dynamic environment.



(a) Coarse Density Process (b) Qualitative Evaluation for Coarse Density

Fig. 2. The loss function (a) for training the coarse network and the rendering results (b) with and without training it are depicted.

from discrete train database. To tackle this problem, two systems are prepared as form of neural networks: a static field (as illustrated in Fig. 1.a) representing the static background, and a transient field (depicted in Fig. 1.b), designed for capturing dynamic motion through learning geometric information.

1) Geometric Field (Fig. 1(a-1)): Prior to every step, we aim to minimize computational latency for adopting to real-world robotics tasks. To achieve this, our method is based on multiresolution hash encoding, proposed Müller et al. [11], which is called h by ours. After hash encoding h, MLP is constructed for extracting the latent space for static color and transient field. The process in Fig. 1(a-1) is as follows, where $\boldsymbol{v}$ is a feature vector extracted by h and $\boldsymbol{\sigma}$, $\boldsymbol{l_s}$, $\boldsymbol{l_t}$ are a static density, latent space of static color and of transient field respectively:

$$\boldsymbol{v} = h(\boldsymbol{x}), \quad (\boldsymbol{\sigma}, \boldsymbol{l_s}, \boldsymbol{l_t}) = M(\boldsymbol{v}) \tag{1}$$

2) Static Color Prediction (Fig. 1(a-2)): This process concentrates on generating static color from viewing direction $(\boldsymbol{\theta}, \boldsymbol{\phi})$ and static latent space $\boldsymbol{l_s}$.

$$\boldsymbol{c^s} = S_c(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{l_s}) \tag{2}$$

3) Transient Field (Fig. 1(b-1)): Similar to static field S, both density $\boldsymbol{\sigma^t}$ and color $\boldsymbol{c^t}$ values are obtained for representing dynamic field D. Additionally, to recognize dynamic objects based on the degree of uncertainty, a 3D uncertainty value $\boldsymbol{\beta} \in \mathbb{R}^1$ is extracted. The uncertainty value $\boldsymbol{\beta}$ is utilized in the loss for decomposition of 3D scenes mentioned in (6).

$$(\boldsymbol{\sigma^t}, \boldsymbol{c^t}, \boldsymbol{\beta}) = T(\boldsymbol{E_t}, \boldsymbol{l_t}) \tag{3}$$

4) Sampling Method focused on Static Field (Fig. 1(a-3)): Photometric-based localization contends with the challenge of high-quality map reconstruction. In order to obtain accurate RGB and depth values, an efficient sampling method is required. We propose a sampling method focused on a static field rather than a transient field to complete dynamic removal, inspired in [12]. The equation is as follows:

$$\mathcal{L}_{prop}(\boldsymbol{t}, \boldsymbol{w}^{s-t}, \boldsymbol{t^c}, \boldsymbol{w^c}) = \sum_i \frac{1}{\omega_i^{s-t}} \max(0, \omega_i^{s-t} - bound(\boldsymbol{t^c}, \boldsymbol{w^c}, T_i))^2 \tag{4}$$

$$\boldsymbol{w}^{s-t} = \max(0, \boldsymbol{w^s} - \boldsymbol{w^t}), bound(\boldsymbol{t^c}, \boldsymbol{w^c}, T) = \sum_{j:T \cap \hat{T}_j \neq \emptyset} \omega_j^c \tag{5}$$

, where $\boldsymbol{t}$, and $\boldsymbol{t}^c$ represent the sampling interval, while $\boldsymbol{w}^{s-t}$, $\boldsymbol{w}^c$ denote the sum of the weights for the static and coarse fields, respectively.

C. Loss Function

Our mapping algorithm is trained with an aleatoric uncertainty loss $\mathcal{L}_{alea}$ and sampling-supervision loss $\mathcal{L}_{prop}$. The aleatoric uncertainty loss (6) utilizes the rendering value of 3D uncertainty, denoted as $\boldsymbol{\beta'}$, to detect dynamics based on the degree of uncertainty.

$$\mathcal{L}_{aleat} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}\exp(-\boldsymbol{\beta'})||\boldsymbol{c_i}-\hat{\boldsymbol{c}}_{\boldsymbol{i}}||^2+\frac{1}{2}\boldsymbol{\beta'}\right] \quad (6)$$

, where $\boldsymbol{c_i}$ and $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ are ground truth and estimated color value (i.e. $\boldsymbol{c_i} = \boldsymbol{c_i^t} + \boldsymbol{c_i^s}$). In summary, the training objective function is as follows:

$$G_t^* = \arg\min_{G_t}\{\lambda_a \cdot \mathcal{L}_{aleat} + \lambda_p \cdot \mathcal{L}_{prop}\} \quad (7)$$

, where λ_a and λ_b are 1.0 and 0.1 in our implementation, respectively.

D. Localization based-on Static Map

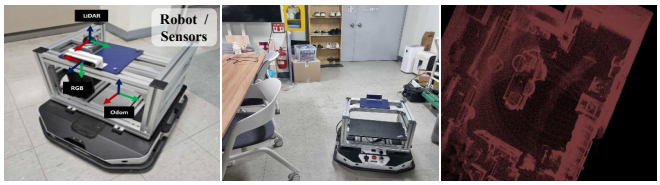
To achieve neural rendering for interaction with robots, a localization task is required, which transcends the literal meaning of rendering. Consequently, our mapping methodology extends beyond the foundation established in Loc-NeRF [13]. Maggio et al. [13] did not account for moving objects in the training phase, where the neural map can deteriorate due to occlusions, leading to diminished localization performance. Moreover, low update frequencies compromise the stability of localization. Therefore, we propose a modified pipeline with increased stability based on multiresolution hash encoding [11] and consideration for particle weighting in the static field S. As depicted in Fig. 1, particles on the neural map are initialized and then propagated based on odometry values. During the particle filter update process, only pixels from static field S are utilized for particle weighting, to avoid the effects of occlusions caused by dynamic objects present at the time of map construction. Particle weighting is conducted according to the following equation:

$$\Omega_i = N\left(\sum_{i=1}^{N}\left(X_t(\pi^i) - S(\theta,\phi,s_{\pi^i})\right)^2\right)^{-M} \quad (8)$$

, where $X_t(\pi^i)$ denotes the pixel value at pixel coordinate π^i in query image, while $S(\theta,\phi,s_{\pi^i})$ denotes the synthesized scene intensity at that coordinate. Ultimately, our pipeline applies a static-aware weight update function to not only perform a dynamic-ignored map but also contribute to more stable robot navigation through higher update frequencies.

III. Experimental Results

A. Experimental Setting



(a) Hardware Setup System (b) Experiment Place (c) Reference Lidar Map

Fig. 3. For our experiments, we utilized a mobile robot equipped with a RGB-D camera (rs-455i) and a MID360 LiDAR for acquiring the reference pose.

The objective of this session is to evaluate the performance of static map reconstruction under real-world scenarios where dynamic objects are present. To achieve this, we employ both open datasets that contain dynamic objects and custom datasets which are generated in an indoor laboratory setting using a mobile robot. The Nvidia RTX 3090 is utilized for training the mapping pipeline, although the system is designed to be scalable to lower-performance devices in robotics, as the network is lightweight.

B. Evaluation: Implicit-based Static Map Reconstruction

1) Quantitative Results: Table I presents the average values of each evaluation on four datasets. For DynamicNeRF, we only show results for rendering static fields because DynamicNeRF(S+T) tended to diverge in loss on our custom dataset. Our method enables the design of networks suitable for robotics tasks with relatively short inference time and high-quality rendering performance.

TABLE I

Quantitative performance comparison for implicit mapping in standard datasets: scores are shown as $mean$; the first and second best scores (in each col) are colored red, and orange, respectively

	PSNR ↑	Abs Err ↓	Train. Time (min) ↓	Infer. Time (sec) ↓
DynamicNeRF(S)	25.20	154.70	23.45	0.124
NSFF	25.03	24.44	160.2	0.247
NeRF	26.26	148.76	55.07	0.514
NeRF-W	26.40	111.45	197.66	0.594
Ours	30.37	7.26	58.72	0.026

2) Qualitative Results: We perform comparative experiments on methodologies that involve reconstructing static fields in datasets containing dynamic objects. To ensure fair training, all models are trained for 20 epochs. As depicted in Fig. 4, NSFF and DynamicNeRF struggle to achieve high rendering performance or complete static map reconstruction on our custom datasets. In contrast, our methodology consistently achieves successful rendering across various datasets, including forward-facing scenes in open datasets and indoor custom and replica datasets. In Fig. 5(a), NeRF-W faces challenges in achieving complete static map reconstruction due to its transient field capturing high-frequency signals. In contrast, our transient field exhibits minimal artifacts. As illustrated in Fig. 5(b), our approach enables high-quality rendering, surpassing NeRF-W, which inaccurately classifies portions of the static environment as dynamic.

C. Evaluation: Localization based on Static Map

In this experiment, we demonstrate that our proposed methodology contributes to the enhancement of localization performance. The comparative model is Loc-NeRF, a particle-filter based localization methodology. Additionally, LiDAR localization is established as the reference pose for our custom dataset.

1) Quantitative Results: To verify the quantitative performance of our localization, we use the Absolute Trajectory Error(ATE). According to Fig. 6, both methodologies show stable performance in a static environment. However when

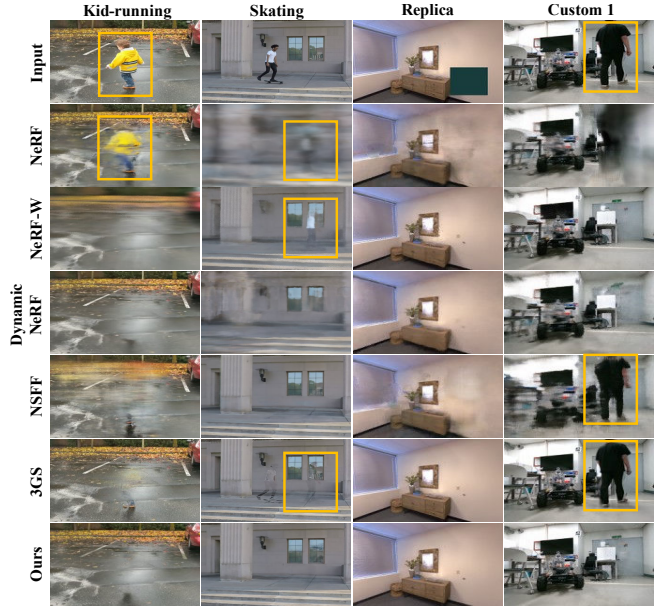
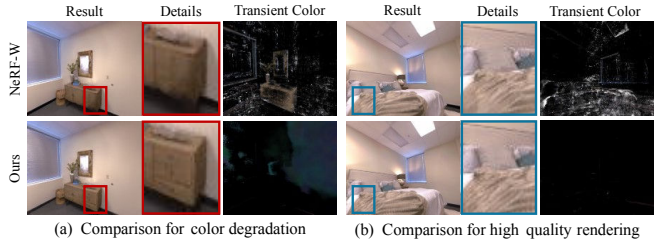


Fig. 4. Qualitative comparison of static implicit mapping on standard and custom datasets.



(a) Comparison for color degradation (b) Comparison for high quality rendering

Fig. 5. Comparative analysis of the qualitative performance in transient color rendering between our method and NeRF-W. NeRF-W's inaccurate transient color estimation diminishes the quality of static map reconstruction task.

it comes to dynamic, the artifacts left in Loc-NeRF's map make localization unstable, whereas our shows decent performance, thanks to the capability of distinguishing static from dynamic scenes. Table II evaluates the quantitative amount of error in position and rotation after convergence. As shown in Fig. 6, our method shows stable performance in dynamic scene with low mean and variance error for position and rotation.

2) Qualitative Results: In the realm of implicit neural map-based localization, filter-based methodologies are significantly affected by the inference time required for particle updates. Fig. 7 presents sequential images rendered from the estimated poses. Since pose estimation relies on analyzing photometric error between the query and rendered images,

TABLE II

COMPARISON OF LOCALIZATION PERFORMANCE ON STATIC AND DYNAMIC ENVIRONMENTS. SCORES REPRESENT POSITION(P)/ROTATION(R) MEAN, VARIANCE ERROR AND POSE UPDATING SPEED.

Method	μ_p (m)	σ_p (cm)	μ_r (rad)	σ_r (rad)	FPS (Hz)
Loc-NeRF (Static)	**0.0453**	2.7222	0.0789	**0.0061**	3.56
Ours (Static)	0.0458	**2.5217**	**0.0495**	0.0110	**5.25**
Loc-NeRF (Dynamic)	0.1099	5.7383	0.3474	0.0140	3.72
Ours (Dynamic)	**0.0645**	**3.4647**	**0.1151**	**0.0086**	**5.03**

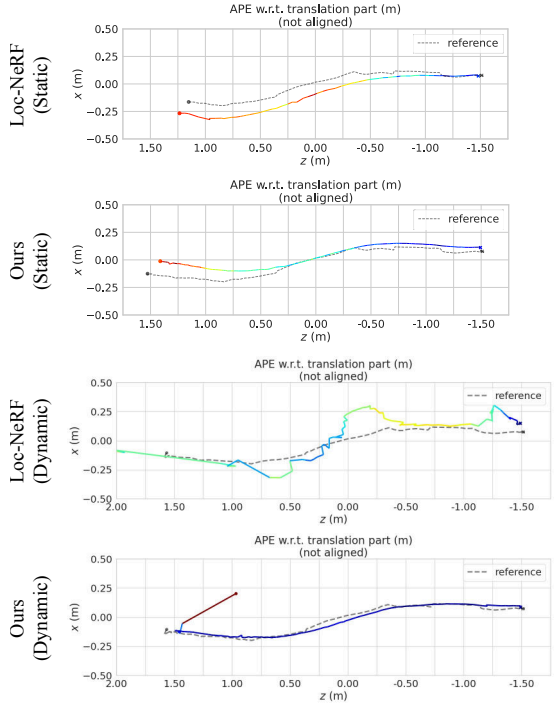


Fig. 6. Localized trajectories by Loc-NeRF and our method, tested under two conditions: with and without dynamic elements. Here, reference denotes LiDAR localization.

any discrepancies observed between these images directly translate into errors in the localization task. Notably, our method has demonstrated superior precision in localization compared to Loc-NeRF. This indicates that the higher update rate of our approach enhances the accuracy of localization.

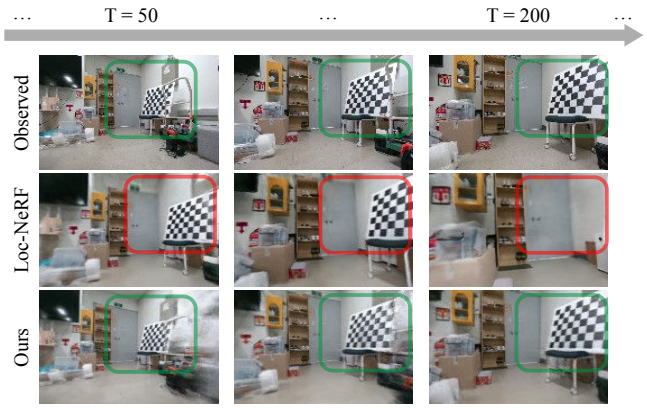


Fig. 7. Rendered images at the poses estimated by each algorithm. Pixel-wise discrepancies between the observed and rendered images indicate localization errors.

IV. CONCLUSION

Our work contributes to the fields of robotics and neural rendering by presenting a static implicit neural map with learning geometric information and then improving the vision-based localization system. We proved that the static map reconstructed by separating occlusion from real dynamic scene play a crucial role in photometric-based robotics tasks. In the future, it could be extended to large-scale, which is limited to about the size of a room, yet.

REFERENCES

[1] 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.

[2] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15 838–15 847.

[3] W. Jang and L. Agapito, "Codenerf: Disentangled neural radiance fields for object categories," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12 949–12 958.

[4] C. Jiang, D. P. Paudel, Y. Fougerolle, D. Fofi, and C. Demonceaux, "Static-map and dynamic object reconstruction in outdoor scenes using 3-d motion segmentation," IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 324–331, 2016.

[5] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments," in 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 3849–3856.

[6] W. Yuan, Z. Lv, T. Schmidt, and S. Lovegrove, "Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13 144–13 152.

[7] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5712–5721.

[8] Z. Li, S. Niklaus, N. Snavely, and O. Wang, "Neural scene flow fields for space-time view synthesis of dynamic scenes," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6498–6508.

[9] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7210–7219.

[10] V. Tschernezki, D. Larlus, and A. Vedaldi, "Neuraldiff: Segmenting 3d objects that move in egocentric videos," in 2021 International Conference on 3D Vision (3DV). IEEE, 2021, pp. 910–919.

[11] 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.

[12] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5470–5479.

[13] D. Maggio, M. Abate, J. Shi, C. Mario, and L. Carlone, "Loc-nerf: Monte carlo localization using neural radiance fields," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 4018–4025.