RoDyn-SLAM: Robust Dynamic Dense RGB-D SLAM with Neural Radiance Fields

Haochen Jiang, Yueming Xu, Kejie Li, Jianfeng Feng, Li Zhang

Abstract—Leveraging neural implicit representation to conduct dense RGB-D SLAM has been studied in recent years. However, this approach relies on a static environment assumption and does not work robustly within a dynamic environment due to the inconsistent observation of geometry and photometry. To address the challenges presented in dynamic environments, we propose a novel dynamic SLAM framework with neural radiance field. Specifically, we introduce a motion mask generation method to filter out the invalid sampled rays. This design effectively fuses the optical flow mask and semantic mask to enhance the precision of motion mask. To further improve the accuracy of pose estimation, we have designed a divide-and-conquer pose optimization algorithm that distinguishes between keyframes and non-keyframes. The proposed edge warp loss can effectively enhance the geometry constraints between adjacent frames. Extensive experiments are conducted on the two challenging datasets, and the results show that RoDyn-SLAM achieves state-of-the-art performance among recent neural RGB-D methods in both accuracy and robustness. Our implementation of the Rodyn-SLAM will be open-sourced to benefit the community[1].

Index Terms—Deep Learning Methods, NeRF, RGB-D SLAM, Dynamic Scene, Pose Estimation.

I. INTRODUCTION

DENSE visual simultaneous localization and mapping (SLAM) is a fundamental task in 3D computer vision and robotics, which has been widely used in various forms in fields such as service robotics, autonomous driving, and augmented/virtual reality (AR/VR). It is defined as reconstructing a dense 3D map in an unknown environment while simultaneously estimating the camera pose, which is regarded as the key to achieving autonomous navigation for robots [1]. However, the majority of methods assume a static environment, limiting the applicability of this technology to more practical scenarios. Thus, it becomes a challenging problem that how the SLAM system can mitigate the interference caused by dynamic objects.

Traditional visual SLAM methods using semantic segmentation prior [2]–[5], optical flow motion [6]–[8] or re-sampling and residual optimization strategies [9]–[11] to remove the outliers under dynamic environments, which can improve the

Haochen Jiang is with the School of Data Science, Fudan University, Shanghai 200433, China. E-mail: jhch1995@mail.ustc.edu.cn. Yueming Xu and Jianfeng Feng are with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai 200433, China. E-mails: xuyueming21@m.fudan.edu.cn, jffeng@fudan.edu.cn. Kejie Li is with ByteDance, Seattle, USA. E-mail: kejie.li@outlook.com.

[1] https://github.com/fudan-zvg/Rodyn-SLAM

accuracy and robustness of pose estimation. However, re-sampling and optimization methods can only handle small-scale motions and often fail when encountering large-scale continuous object movements. Moreover, semantic priors are specific to particular categories and can not represent the real motion state of the observation object. The above learning-based methods often exhibit a domain gap when applied in real-world environments, leading to the introduction of prediction errors.

Recently, dense visual SLAM with neural implicit representation has gained more attention and popularity. This novel map representation is more compact, continuous, efficient, and able to be optimized with differentiable rendering, which has the potential to benefit applications like navigation, planning, and reconstruction. Moreover, the neural scene representations have attractive properties for mapping, including improving noise and outlier handling, geometry estimation capabilities for unobserved scene parts, high-fidelity reconstructions with reduced memory usage, and the ability to generate high-quality static background images from novel views. Existing methods like iMap [12] and NICE-SLAM [13] respectively leverage single MLP and hierarchical feature grids to achieve a consistent geometry representation. However, these methods have limited capacity to capture intricate geometric details. Recent works such as Co-SLAM [14] and ESLAM [15] explore hash encoding or tri-plane representation strategy to enhance the capability of scene representation and the system's execution efficiency. However, all these above-mentioned methods do not perform well in dynamic scenes. The robustness of these systems significantly decreases, even leading to tracking failures when dynamic objects appear in the environment.

To tackle these problems, we propose a novel NeRF-based RGB-D SLAM that can reliably track camera motion in indoor dynamic environments. One of the key elements to improve the robustness of pose estimation is the motion mask generation algorithm that filters out the sampled rays located in invalid regions. By incrementally fusing the optical flow mask [16], the semantic segmentation mask [17] can become more precise to reflect the true motion state of objects. To further improve the accuracy of pose estimation, we design a divide-and-conquer pose optimization algorithm for keyframes and non-keyframes. While an efficient edge warp loss is used to track camera motions for all keyframes and non-keyframes w.r.t. adjacent frames, only keyframes are further jointly optimized via rendering loss in the global bundle adjustment (GBA).

In summary, our **contributions** are summarized as follows:
1) To the best of our knowledge, this is the first dynamic neural RGB-D SLAM with joint robust pose estimation

and dense reconstruction.

2) In response to the issue of inaccurate semantic priors, we propose a motion mask generation strategy fusing spatial-temporal consistent optical flow masks to improve the robustness of camera pose estimation and quality of static scene reconstruction.

3) Instead of a single frame tracking method, we design a novel mixture pose optimization algorithm utilizing an edge warp loss to enhance the geometry consistency in the non-keyframe tracking stage.

4) We evaluate our method on two challenging dynamic datasets to demonstrate the state-of-the-art performance of our method in comparison to existing NeRF-based RGB-D SLAM approaches.

II. RELATED WORK

A. Conventional visual SLAM with dynamic objects filter

Dynamic object filtering aims to reconstruct the static scene and enhance the robustness of pose estimation. Prior methods can be categorized into two groups: the first one utilizes the re-sampling and residual optimization strategies to remove the outliers [9]–[11]. However, these methods can only handle small-scale motions and often fail when encountering large-scale continuous object movements. The second group employs the additional prior knowledge, such as semantic segmentation prior [2]–[5], [18] or optical flow motion [6]–[8] to remove the dynamic objects. However, all these methods often exhibit a domain gap when applied in real-world environments, leading to the introduction of prediction errors. In this paper, we propose a motion mask generation strategy that complements the semantic segmentation mask with warping optical flow masks [16], [19], which is beneficial for reconstructing more accurate static scene maps and reducing observation error.

B. RGB-D SLAM with neural implicit representation

Neural implicit scene representations, also known as neural fields [20], have garnered significant interest in RGB-D SLAM due to their expressive capacity and minimal memory requirements. iMap [12] firstly adopts a single MLP representation to jointly optimize camera pose and implicit map throughout the tracking and mapping stages. However, it suffers from representation forgetting problems and fails to produce detailed scene geometry. DI-Fusion [21] encodes the scene prior in a latent space and optimizes a feature grid, but it leads to poor reconstruction quality replete with holes. NICE-SLAM [13] leverages a multi-level feature grid enhancing scene representation fidelity and utilizes a local feature update strategy to reduce network forgetting. However, it remains memory-intensive and lacks real-time capability. More recently, existing methods like Vox-Fusion [22], Co-SLAM [14], and ESLAM [15] explore sparse encoding or tri-plane representation strategy to improve the quality of scene reconstruction and the system's execution efficiency. All these methods have demonstrated impressive results based on the strong assumptions of static scene conditions. The robustness of these systems significantly decreases when dynamic objects

appear in the environment. Our SLAM system aims to enhance the accuracy and robustness of pose estimation under dynamic environments, which can expand the application range for the NeRF-based RGB-D SLAM system.

C. Dynamic objects decomposition in NeRFs

As the field of NeRF continues to advance, some researchers are attempting to address the problem of novel view synthesis in the presence of dynamic objects. One kind of solution is to decompose the static background and dynamic objects with different neural radiance fields like [23]–[29]. The time dimension will be encoded in latent space, and novel view synthesis is conducted in canonical space. Although these space-time synthesis results are impressive, these techniques rely on precise camera pose input. Robust-Dynrf [30] jointly estimate the static and dynamic radiance fields along with the camera parameters (poses and focal length), which can achieve the unknown camera pose training. However, it can not directly apply to RGB-D SLAM system for large-scale tracking and mapping. Another kind of solution is to ignore the dynamic objects' influence by utilizing robust loss and optical flow like [28], [31]. Compared to the dynamic NeRF problem, we often focus on the accuracy of pose estimation and the quality of static reconstruction without a long training period. Thus, we also ignore modeling dynamic objects and propose a robust loss function with a novel optimization strategy to recover the static scene map.

III. METHOD

Given a sequence of RGB-D frames $\{I_i, D_i\}_{i=1}^{N}, I_i \in \mathbb{R}^3, D_i \in \mathbb{R}$, our method (Fig. 1) aims to simultaneously recover camera poses $\{\xi_i\}_{i=1}^{N}, \xi_t \in \mathbb{SE}(3)$ and reconstruct the static 3D scene map represented by neural radiance fields in dynamic environments. Similar to most modern SLAM systems [32], [33], our system comprises two distinct processes: the tracking process as the frontend and the mapping process as the backend, combined with keyframe management $\{F_k\}_{k=1}^{M}$ and neural implicit map f_θ. Invalid sampling rays within dynamic objects are filtered out using a motion mask generation approach. Contrary to the conventional constant-speed motion model in most systems, we introduce an edge warp loss for optimization in non-keyframes to enhance the robustness of pose estimation. Furthermore, keyframe poses and the implicit map representations are iteratively optimized using differentiable rendering.

A. Implicit map representation

We introduce two components of our implicit map representation: an efficient multi-resolution hash encoding $\mathbb{V}_\alpha$ to encode the geometric information of the scene, and individual tiny MLP decoders f_ϕ to render the color and depth information with truncated signed distance (TSDF) prediction.

a) Multi-resolution hash encoding: We use a multi resolution hash-based feature grid $\mathbb{V}_\alpha = \{V_\alpha^l\}_{l=1}^{L}$ and individual shallow MLPs to represent the implicit map following Instant-NGP [34]. The spatial resolution of each level is progressively
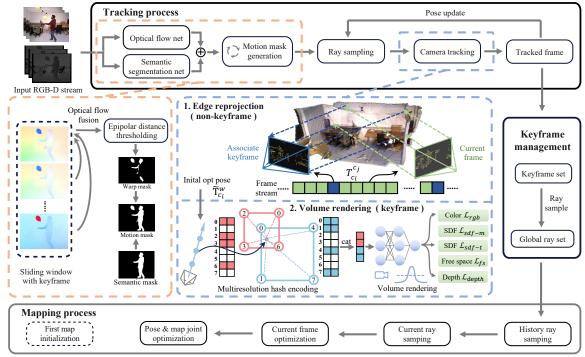
Fig. 1. **The schematic illustration of the proposed method.** Given a series of RGB-D frames, we simultaneously construct the implicit map and camera pose via multi-resolution hash gird with the geometric loss $\mathcal{L}_{sdf\text{-}m}, \mathcal{L}_{sdf\text{-}t}, \mathcal{L}_{fs}, \mathcal{L}_{depth}$, color loss $\mathcal{L}_{color}$, and edge warp loss $\mathcal{L}_{edge}$.

set between the coarsest resolution, denoted as R_{min}, and the finest resolution, represented as R_{max}. Given a sampled point $\mathbf{x}$ in 3D space, we compute the interpolate feature $V_\alpha^l(\mathbf{x})$ from each level via trilinear interpolation. To obtain more complementary geometric information, we concat the encoding features from all levels as the input of the MLPs decoder. While simple MLPs can lead to the issue of catastrophic forgetting [12], [13], this **mechanism of forgetfulness** can be leveraged to eliminate historical dynamic objects.

b) Color and depth rendering: To obtain the final formulation of implicit map representation, we adopt a two-layer shallow MLP to predict the geometric and appearance information, respectively. The geometry decoder outputs the predicted SDF value s and a feature vector $\mathbf{h}$ at the point $\mathbf{x}$. The appearance decoder outputs the predicted RGB value c. Similar to Co-SLAM [14], we joint encode the coordinate encoding $\gamma(\mathbf{x})$ and parametric encoding V_α as:

$$f_\beta\left(\gamma(\mathbf{x}), V_\alpha(\mathbf{x})\right) \mapsto (\mathbf{h}, s), \quad f_\phi\left(\gamma(\mathbf{x}), \mathbf{h}\right) \mapsto \mathbf{c}, \quad (1)$$

where $\{\alpha, \beta, \phi\}$ are the learnable parameters. Following the volume rendering method in NeRF [35], we accumulate the predicted values along the viewing ray $\mathbf{r}$ at the current estimation pose ξ_i to render the color and depth value as:

$$\hat{C}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i \mathbf{c}_i, \quad \hat{D}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i z_i, \quad (2)$$

where w_i is the computed weight along the ray, $\mathbf{c}_i$ and z_i are the color and depth value of the sampling point $\mathbf{x}_i$. Since we do not directly predict voxel density σ like NeRF, here we need to convert the SDF values s_i into weights w_i. Thus, we employ a straightforward bell-shaped function [36], formulated as the

product of two sigmoid functions $\sigma(\cdot)$.

$$w_i = \sigma\left(\frac{s_i}{tr}\right) \sigma\left(-\frac{s_i}{tr}\right), \quad \hat{D}_{var}(\mathbf{r}) = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i(\hat{D} - z_i)^2, \quad (3)$$

where tr denotes the truncation distance with TSDF prediction, $\hat{D}_{var}$ is the depth variance along this ray. When possessing GT depth values, we opt for uniform point sampling near the surface rather than employing importance sampling, with the aim of enhancing the efficiency of point sampling.

B. Motion mask generation

For each input keyframe, we select its associated keyframes within a sliding window to compute the dense optical flow warping set $\mathcal{S}$. Note that optical flow estimation is conducted solely on keyframes, thereby optimizing system efficiency. To separate the ego-motion from dynamic objects, we additionally estimate the fundamental matrix $\boldsymbol{F}$ with inliers sampled from the matching set $\mathcal{S}$. Given any matching points $\boldsymbol{o}_{ji}, \boldsymbol{o}_{ki}$ within $\mathcal{S}$, we utilize matrix $\boldsymbol{F}$ to compute the Sampson distance between corresponding points and their epipolar lines. By setting a suitable threshold e_{th}, we derive the warp mask $\widehat{\mathcal{M}}_{j,k}^{wf}$ corresponding to dynamic objects as:

$$\widehat{\mathcal{M}}_{j,k}^{wf} : \left\{ \bigcap_{i=1}^{M} \mathbf{1}\left(\frac{\boldsymbol{o}_{ji}^T \boldsymbol{F} \boldsymbol{o}_{ki}}{\sqrt{A^2 + B^2}} < e_{th}\right) \otimes \boldsymbol{I}_{m \times n} \,\middle|\, \forall (\boldsymbol{o}_{ji}, \boldsymbol{o}_{ki}) \in \mathcal{S} \right\} \quad (4)$$

where A, B denotes the coefficients of the epipolar line, and m, n represents the size of the warp mask, aligning with the current frame image's dimensions. Additionally, j and k stand for the keyframe ID, illustrating the optical flow mask warping process from the k-th to the j-th keyframe. As illustrated in Fig. 1, to derive a more precise motion mask, we consider the spatial coherence of dynamic object motions

within a sliding window of length N and iteratively optimize the current motion mask. Subsequently, we integrate the warp mask and segment mask to derive the final motion mask $\widehat{\mathcal{M}}_j$ as:

$$\widehat{\mathcal{M}}_j = \widehat{\mathcal{M}}_{j,k}^{wf} \otimes \widehat{\mathcal{M}}_{j,k-1}^{wf} \otimes \widehat{\mathcal{M}}_{j,k-2}^{wf} \cdots \otimes \widehat{\mathcal{M}}_{j,k-N}^{wf} \cup \widehat{\mathcal{M}}_j^{sg}, \tag{5}$$

where $\otimes$ represents the mask fusion operation, which is applied when pixels corresponding to a specific motion mask have been continuously observed for a duration exceeding a certain threshold o_{th} within a sliding window. Note that we do not focus on the specific structure of the segment or optical flow network. Instead, we aim to introduce a general motion mask fusing method for application in NeRF-based SLAMs. We believe that there is potential for integrating this approach into any visual SLAM system.

C. Joint optimization

We introduce the details on optimizing the implicit scene representation and camera pose. Given a set of frames $\mathcal{F}$, we only predict the current camera pose represented with lie algebra ξ_i in tracking process. Moreover, we utilize the global bundle adjustment (GBA) [37]–[39] to jointly optimize the sampled camera pose and the implicit mapping.

1) Photometric rendering loss: To jointly optimize the scene representation and camera pose, we render depth and color in independent view as Eq. 6 comparing with the proposed ground truth map:

$$\mathcal{L}_{rgb} = \frac{1}{M} \sum_{i=1}^{M} \left\| \left(\hat{C}(\mathbf{r}) - C(\mathbf{r}) \right) \cdot \widehat{\mathcal{M}}_i(\mathbf{r}) \right\|_2^2,$$

$$\mathcal{L}_{depth} = \frac{1}{N_d} \sum_{\mathbf{r} \in N_d} \left\| \left(\frac{\hat{D}(\mathbf{r}) - D(\mathbf{r})}{\sqrt{\hat{D}_{var}(\mathbf{r})}} \right) \cdot \widehat{\mathcal{M}}_i(\mathbf{r}) \right\|_2^2, \tag{6}$$

where $C(\mathbf{r})$ and $D(\mathbf{r})$ denote the ground truth color and depth map corresponding with the given pose,s respectively. M represents the number of sampled pixels in the current image. Note that only rays with valid depth value N_d are considered in $\mathcal{L}_{depth}$. In contrast to existing methods, we introduce the motion mask $\widehat{\mathcal{M}}_j$ to remove sampled pixels within the dynamic object region effectively. Moreover, to improve the robustness of pose estimation, we add the depth variance $\hat{D}_{var}$ to reduce the weight of depth outliers.

2) Geometric constraints: Following the practice [36], assuming a batch of rays M within valid motion mask regions are sampled, we directly leverage the free space loss with truncation tr to restrict the SDF values $s(\mathbf{x_i})$ as:

$$\mathcal{L}_{fs} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\mathcal{R}_{fs}|} \sum_{i \in \mathcal{R}_{fs}} (s(\mathbf{x_i}) - tr)^2, \quad [u_i, v_i] \subseteq (\widehat{\mathcal{M}}_i = 1). \tag{7}$$

It is unreasonable to employ a fixed truncation value to optimize camera pose and SDF values in dynamic environments simultaneously. To reduce the artifacts in occluded areas and enhance the accuracy of reconstruction, we further divide the

entire truncation region near the surface into middle and tail truncation regions inspired by ESLAM [15] as:

$$\mathcal{L}_{sdf} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\mathcal{R}_{tr}|} \sum_{i \in \mathcal{R}_{tr}} (s(\mathbf{x}_i) - (D[u_i, v_i] - T \cdot tr))^2, \tag{8}$$

where T denotes the ratio of the entire truncation length occupied by the middle truncation, $[u_i, v_i] \subseteq (\widehat{\mathcal{M}}_i = 1)$. Note that we use the different weights to adjust the importance of middle and tail truncation in camera tracking and mapping process. The overall loss function is finally formulated as the following minimization,

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \lambda_1 \mathcal{L}_{rgb} + \lambda_2 \mathcal{L}_{depth} + \lambda_3 \mathcal{L}_{fs} + \lambda_4 \mathcal{L}_{sdf\text{-}m} + \lambda_5 \mathcal{L}_{sdf\text{-}t}, \tag{9}$$

where $\mathcal{P} = \{\theta, \phi, \alpha, \beta, \gamma, \xi_i\}$ is the list of parameters being optimized, including fields feature, decoders, and camera pose.

3) Camera tracking process: The construction of implicit maps within dynamic scenes often encounters substantial noise and frequently exhibits a lack of global consistency. Existing methods [13]–[15], [40] rely solely on rendering loss for camera pose optimization, which makes the system vulnerable and prone to tracking failures. To solve this problem, we introduce edge warp loss to enhance geometry consistency in data association between adjacent frames.

Edge reprojection loss. For a 2D pixel p in frame i, we first define the warp operation in a similar spirit as DIM-SLAM [40] to reproject it onto frame j as follows:

$$\boldsymbol{p}_{i \to j} = f_{warp}\left(\xi_{ji}, \boldsymbol{p}_i, D(\boldsymbol{p}_i)\right) = \boldsymbol{K}\boldsymbol{T}_{ji}\left(\boldsymbol{K}^{-1}D(\boldsymbol{p}_i)\boldsymbol{p}_i^{homo}\right), \tag{10}$$

where $\boldsymbol{K}$ and $\boldsymbol{T}_{ji}$ represent the intrinsic matrix and the transformation matrix between frame i and frame j, respectively. $\boldsymbol{p}_i^{homo} = (u, v, 1)$ is the homogeneous coordinate of $\boldsymbol{p}_i$. Since the edge are detected once and do not change forwards, we can precompute the distance map (DT) [41] to describe the projection error with the closest edge. For a edge set $\mathcal{E}_i$ in frame i, we define the edge loss $\mathcal{L}_{edge}$ as

$$\mathcal{L}_{edge} = \sum_{\boldsymbol{p}_i \in \mathcal{E}_i} \rho(\mathcal{D}_j(f_{warp}\left(\xi_{ji}, \boldsymbol{p}_i, D(\boldsymbol{p}_i)\right)) \cdot \widehat{\mathcal{M}}_j), \tag{11}$$

where $\mathcal{D}_j$ denotes the DT map in frame j, and the ρ is a Huber weight function to reduce the influence of large residuals. Moreover, we drop a potential outlier if the projection distance error is greater than δ_e. The pose optimization problem is finally formulated as the following minimization,

$$\xi_{ji}^* = \arg\min_{\xi_{ji}} \lambda \mathcal{L}_{edge}, \quad \text{if } j \notin \mathcal{K} \tag{12}$$

To further improve the accuracy and stability of pose estimation, we employ distinct methods for tracking keyframes and non-keyframes in dynamic scenes. Keyframe pose estimation utilizes the edge loss to establish the initial pose, followed by optimization (Eq. 9). For non-keyframe pose estimation, we optimize the current frame's pose related to the nearest keyframe (Eq. 12).
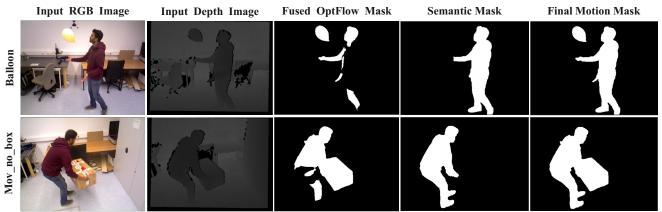
Fig. 2. **Qualitative results of the generation motion mask.** By iteratively optimizing the optical flow mask, the fused optical mask can be more precise without noises. The semantic mask can only identify dynamic objects within predefined categories. The best results are obtained with our method.

IV. EXPERIMENTS

Datasets. We evaluate our method on two real-world public datasets: *TUM RGB-D* dataset [42] and *BONN RGB-D Dynamic* dataset [11]. Both datasets capture indoor scenes using a handheld camera and provide the ground-truth trajectory.

Metrics. For evaluating pose estimation, we adopt the RMSE and STD of Absolute Trajectory Error (ATE) [42]. The estimated trajectory is oriented to align with the ground truth trajectory using the unit quaternions algorithm [43] before evaluation. We also use three metrics which are widely used for scene reconstruction evaluation following [13], [40]: (i) *Accuracy* (cm), (ii) *Completion* (cm), (iii) *Completion Ratio* (< 5cm %). Since the BONN-RGBD only provided the ground truth point cloud, we randomly sampled the 200,000 points from both the ground truth point cloud and the reconstructed mesh surface to compute the metrics. We remove unobserved regions that are outside of any camera's viewing frustum and conduct extra mesh culling to remove the noisy points external to the target scene [14].

Implementation details. We adopt Co-SLAM [14] as the baseline in our experiments and run our RoDyn-SLAM on an high-performance workstation with a 3.4GHz Intel Core i7-13700K CPU and RTX 3090Ti GPU at 10 FPS (without optical flow mask) on the Tum datasets, which takes roughly 4GB of memory in total. Specific to implementation details, we sample $N_t = 1024$ rays and $N_p = 85$ points along each camera ray with 20 iterations for tracking and 2048 pixels from every 5 th frames for global bundle adjustment. We set loss weight $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, $\lambda_3 = 10$, $\lambda_4 = 2000$, $\lambda_5 = 500$ to train our model with Adam [44] optimizer. In the motion mask generation method, we utilize Oneformer [17] for semantic segmentation prior generation and RAFT-GMA [16] for optical flow prediction. In the edge extraction process, we utilize the Canny [45] edge detection algorithm with double-threshold. For the sake of comparison fairness, we employ the same keyframe insertion strategy as Co-SLAM [14].

A. Evaluation of generating motion mask

Fig. 2 shows the qualitative results of the generated motion mask. We evaluated our method on the `balloon` and `move_no_box2` sequence of the *BONN* dataset. In these sequences, in addition to the movement of the person, there are

Fig. 3. **Visual comparison of the reconstructed meshes on the *BONN* and *TUM RGB-D* datasets.** Our results are more complete and accurate without the dynamic object floaters.

also other dynamic objects associated with the person, such as balloons and boxes. As shown in Fig. 2 final mask part, our methods can significantly improve the accuracy of motion mask segmentation and effectively mitigate both false positives and false negatives issues in motion segmentation.

B. Evaluation of mapping and tracking

a) Mapping: To better demonstrate the performance of our proposed system in dynamic scenes, we evaluate the mapping results from both qualitative and quantitative perspectives. Since the majority of dynamic scene datasets do not provide ground truth for static scene reconstruction, we adopt the *BONN* dataset to conduct quantitative analysis experiments. We compare our RoDyn-SLAM method against traditional dynamic SLAM method like ReFusion [11] and current state-of-the-art NeRF-based methods with RGB-D sensors, including NICE-SLAM [13], iMap [12], Vox-Fusion [22], ESLAM [15], and Co-SLAM [14], which are open source. The evaluation metrics have been mentioned above at the beginning of Section IV.

As shown in Tab. I, our method outperforms most of the neural RGB-D slam systems on accuracy and completion. To improve the accuracy of pose estimation, we filter the invalid depth, which may reduce the accuracy metric on mapping evaluation. The visual comparison of reconstructed meshes

TABLE I
QUANTITATIVE RESULTS ON SEVERAL DYNAMIC SCENE SEQUENCES IN THE *BONN-RGBD* DATASET. "X" DENOTES THE TRACKING FAILURES. THE BEST RESULTS ARE **BOLDED**, AND THE SECOND BEST RESULTS ARE INDICATED WITH AN UNDERLINE.

		ball	ball2	ps_trk	ps_trk2	mv_box2	Avg.
ReFusion [11]	**Acc.**[cm]↓	**8.20**	**7.85**	46.89	78.47	**9.07**	30.10
	Comp.[cm]↓	12.58	11.69	104.04	166.63	13.09	61.61
	Comp. Ratio[≤ 5cm%]↑	31.57	32.18	13.93	10.55	35.51	24.75
iMAP* [12]	**Acc.**[cm]↓	16.68	31.20	35.38	54.16	17.01	30.89
	Comp.[cm]↓	27.32	30.14	201.38	107.28	20.499	77.32
	Comp. Ratio[≤ 5cm%]↑	25.68	21.91	11.54	12.63	24.86	19.32
NICE-SLAM [13]	**Acc.**[cm]↓	X	24.30	43.11	74.92	17.56	39.97
	Comp.[cm]↓	X	16.65	117.95	172.20	18.19	81.25
	Comp. Ratio[≤ 5cm%]↑	X	29.68	15.89	13.96	32.18	22.93
Vox-Fusion [22]	**Acc.**[cm]↓	85.70	89.27	208.03	162.61	40.64	117.25
	Comp.[cm]↓	55.01	29.78	279.42	229.79	28.40	124.48
	Comp. Ratio[≤ 5cm%]↑	3.88	11.76	2.17	4.55	14.69	7.41
Co-SLAM [14]	**Acc.**[cm]↓	10.61	14.49	26.46	26.00	12.73	18.06
	Comp.[cm]↓	10.65	40.23	124.86	118.35	10.22	60.86
	Comp. Ratio[≤ 5cm%]↑	34.10	3.21	2.05	2.90	39.10	16.27
ESLAM [15]	**Acc.**[cm]↓	17.17	26.82	59.18	89.22	12.32	40.94
	Comp.[cm]↓	9.11	13.58	145.78	186.65	10.03	73.03
	Comp. Ratio[≤ 5cm%]↑	47.44	**47.94**	20.53	17.33	41.41	34.93
Ours(RoDyn-SLAM)	**Acc.**[cm]↓	10.60	13.36	**10.21**	**13.77**	11.34	**11.86**
	Comp.[cm]↓	**7.15**	**7.87**	**27.70**	**18.97**	**6.86**	**13.71**
	Comp. Ratio[≤ 5cm%]↑	**47.58**	40.91	**34.13**	**32.59**	**45.37**	**40.12**

TABLE II
CAMERA TRACKING RESULTS ON SEVERAL DYNAMIC AND STATIC SCENE SEQUENCES IN THE *TUM RGB-D* DATASET. "∗" DENOTES THE VERSION REPRODUCED BY NICE-SLAM. "-" DENOTE THE ABSENCE OF MENTION. THE METRIC UNIT IS [CM].

Methods	Dense	Dynamic								Static				Avg.	
		f3/wk_xyz		f3/wk_hf		f3/wk_st		f3/st_hf		f1/xyz		f1/rpy			
Traditional SLAM methods	T/F	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.
ORB-SLAM3 [10]	✗	28.1	12.2	30.5	9.0	2.0	1.1	**2.6**	1.6	**1.1**	0.6	2.2	1.3	11.1	4.3
DVO-SLAM [46]	✓	59.7	-	52.9	-	21.2	-	6.2	-	**1.1**	-	**2.0**	-	22.9	-
DynaSLAM [3]	✗	**1.7**	-	**2.6**	-	**0.7**	-	2.8	-	-	-	-	-	**2.0**	-
ReFusion [11]	✓	9.9	-	10.4	-	1.7	-	11.0	-	-	-	-	-	8.3	-
NeRF based SLAM methods	T/F	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.
iMAP* [12]	✓	111.5	43.9	X	X	137.3	21.7	93.0	35.3	7.9	7.3	16.0	13.8	73.2	24.4
NICE-SLAM [13]	✓	113.8	42.9	X	X	88.2	27.8	45.0	14.4	4.6	3.8	3.4	2.5	51	18.3
Vox-Fusion [22]	✓	146.6	32.1	X	X	109.9	25.5	89.1	28.5	1.8	0.9	4.3	3.0	70.4	18
Co-SLAM [14]	✓	51.8	25.3	105.1	42.0	49.5	10.8	4.7	2.2	2.3	1.2	3.9	2.8	36.3	14.1
ESLAM [15]	✓	45.7	28.5	60.8	27.9	93.6	20.7	**3.6**	**1.6**	**1.1**	0.6	**2.2**	**1.2**	34.5	13.5
RoDyn-SLAM(Ours)	✓	**8.3**	**5.5**	**5.6**	**2.8**	**1.7**	**0.9**	4.4	2.2	1.5	0.8	2.8	1.5	**4.1**	**2.3**

with other methods [14], [15] is provided in Fig. 3. Note that the TUM dataset does not provide ground truth meshes for evaluating mapping quality. Our methods can generate a more accurate static mesh than other compared methods. Since the baseline methods [14] adopt the hash encoding to represent the implicit map, it may exacerbate the issue of the hash collisions in dynamic scenes and generate the hole in the reconstruction map.

b) Tracking: To evaluate the accuracy of camera tracking in dynamic scenes, we compare our methods with the recent neural RGB-D SLAM methods and traditional SLAM methods like ORB-SLAM3 [10], DVO-SLAM [46], Droid-SLAM [47], and traditional dynamic SLAM like DynaSLAM [3], and ReFusion [11].

As shown in Tab. II, we report the results on three highly dynamic sequences, one slightly dynamic sequence, and two static sequences from TUM RGB-D dataset. Our system achieves advanced tracking performance owing to the motion mask filter and edge-based optimization algorithm under dynamic environment. Compared with our baseline methods Co-SLAM [14], our method does not compromise the performance of the original SLAM methods in terms of tracking and mapping in static scenes. In fact, it achieves competitive results. Notably, our proposed optimization algorithm is not restricted to a specific slam system. Thus, it can also be applied to other neural rgb-d slam methods to improve the data association between the inter-frame. We have also evaluated the tracking performance on the more complex and challenging BONN RGB-D dataset, as illustrated in Tab. III. In more complex and challenging scenarios, our method has achieved superior results. While there is still some gap compared to the more mature and robust traditional dynamic SLAM methods, our systems can drive the dense and textural reconstruction map to finish the more complex robotic navigation tasks.

TABLE III
CAMERA TRACKING RESULTS ON SEVERAL DYNAMIC SCENE SEQUENCES IN THE *BONN RGB-D* DATASET. "∗" DENOTES THE VERSION REPRODUCED BY NICE-SLAM. "-" DENOTE THE ABSENCE OF MENTION, RESPECTIVELY. THE METRIC UNIT IS [CM].

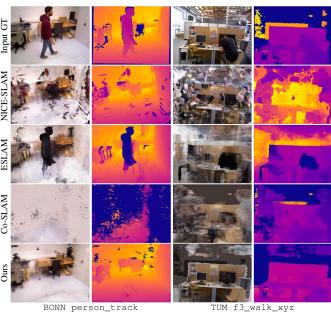| Methods	Dense	balloon ATE	S.D.	balloon2 ATE	S.D.	ps_track ATE	S.D.	ps_track2 ATE	S.D.	ball_track ATE	S.D.	mv_box2 ATE	S.D.	Avg. ATE	S.D.
Traditional SLAM methods	T/F	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.
ORB-SLAM3 [10]	✗	5.8	2.8	17.7	8.6	70.7	32.6	77.9	43.8	**3.1**	1.6	**3.5**	1.5	29.8	15.2
Droid-VO [47]	✓	5.4	-	4.6	-	21.34	-	46.0	-	8.9	-	5.9	-	15.4	-
DynaSLAM [3]	✗	**3.0**	-	**2.9**	-	**6.1**	-	**7.8**	-	4.9	-	3.9	-	**4.8**	-
ReFusion [11]	✓	17.5	-	25.4	-	28.9	-	46.3	-	30.2	-	17.9	-	27.7	-
NeRF based SLAM methods	T/F	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.	ATE	S.D.
iMAP* [12]	✓	14.9	5.4	67.0	19.2	28.3	12.9	52.8	20.9	24.8	11.2	28.3	35.3	36.1	17.5
NICE-SLAM [13]	✓	X	X	66.8	20.0	54.9	27.5	45.3	17.5	21.2	13.1	31.9	13.6	44.1	18.4
Vox-Fusion [22]	✓	65.7	30.9	82.1	52.0	128.6	52.5	162.2	46.2	43.9	16.5	47.5	19.5	88.4	36.3
Co-SLAM [14]	✓	28.8	9.6	20.6	8.1	61.0	22.2	59.1	24.0	38.3	17.4	70.0	25.5	46.3	17.8
ESLAM [15]	✓	22.6	12.2	36.2	19.9	48.0	18.7	51.4	23.2	**12.4**	6.6	17.7	7.5	31.4	14.7
RoDyn-SLAM(Ours)	✓	**7.9**	2.7	**11.5**	6.1	**14.5**	4.6	**13.8**	3.5	13.3	4.7	**12.6**	4.7	**12.3**	4.4

Fig. 4. **Visual comparison of the rendering image on the *TUM* and *BONN* datasets.**

C. Ablation study

To demonstrate the effectiveness of the proposed methods in our system, we perform the ablation studies on seven representative sequences of the *BONN* dataset, including `person_tracking`, `balloon`, `balloon_track`, `move_no_box`. As the semantic prior in the TUM dataset already covers most of the motion categories, we did not conduct ablation studies on this dataset. We compute the average ATE and STD results to show how different methods affect the overall system performance. The results presented in Tab. IV demonstrate that all the proposed methods are effective in camera tracking. This suggests that fusing the optical flow mask and semantic motion mask can promote better pose estimation. At the same time, leveraging a divide-and-conquer pose optimization can effectively improve the robustness and accuracy of camera tracking.

D. Time consumption analysis

As shown in Tab. V, we report time consumption (per frame) of the tracking and mapping without computing se-

TABLE IV
ABLATION STUDY OF THE PROPOSED METHOD IN OUR SYSTEMS.

	w/o Seg	w/o Flow	w/o Edge	RoDyn-SLAM
ATE RMSE (m) ↓	0.3089	0.1793	0.2056	**0.1354**
STD (m) ↓	0.1160	0.0739	0.0829	**0.0543**

TABLE V
TIME COMPARISON OF DIFFERENT METHODS IN OUR SYSTEMS.

	NICE-SLAM	ESLAM	Co-SLAM	RoDyn-SLAM
Tracking (ms) ↓	3535.67	1002.52	174.47	**159.06**
Mapping (ms) ↓	3055.58	703.69	**565.50**	675.08

mantic segmentation and optical flow. Note that we pay more attention to evaluating the impact of our proposed methods on the baseline SLAM system's runtime. All the results were obtained using an experimental configuration of sampled 1024 pixels and 20 iterations for tracking and 2048 pixels and 40 iterations for mapping, with an RTX 3090 GPU in our laboratory server. Despite incorporating additional methods for handling dynamic objects, our system maintains a comparable level of computational cost to that of Co-SLAM. We also evaluate the time efficiency of our used optical flow and semantic segmentation network in our laboratory server, which required 97ms and 163ms respectively to process a single frame. Since semantic segmentation results can be pre-generated, the overall execution time of our optical flow fusion module is approximately 247ms. Note that Rodyn-SLAM is not optimized for real-time operation. With ongoing advancements in these research fields and improvements in computing power, the processing speeds for optical flow and semantic segmentation are expected to increase, ensuring they do not become bottlenecks for our method.

E. Visualization of Rendering Static Implicit Map

To further demonstrate the performance of static scene reconstruction, we compared the rendered image with the ground truth pose obtained from the generated static implicit map. We selected two challenging sequences, `person_track` from the BONN dataset and `f3_walk_xyz` from the TUM RGB-D dataset. As shown in Fig. 4, our method achieves a favorable rendering performance while enjoying the benefits of the proposed methods. Meanwhile, our methods can fill

the hole which can not be captured in the original depth image. It can make the scene representation smoother and more complementing. We observed variations in rendering capabilities among different methods, which resulted in differences in the presentation quality. Note that our methods can be incrementally implemented in any existing baseline methods. Therefore, we don't focus on the actual performance of the code base Co-SLAM [14] but solely on the proposed methods's ability and effectiveness in addressing dynamic scene challenges.

V. CONCLUSION

We present RoDyn-SLAM, a novel dense RGB-D SLAM with neural implicit representation for dynamic environments. The proposed system is able to estimate camera poses and recover 3D geometry in this challenging setup thanks to the motion mask generation that successfully filters out dynamic regions. To further improve the stability and robustness of pose optimization, a divide-and-conquer pose optimization algorithm is designed to enhance the geometry consistency between keyframe and non-keyframe with the edge warp loss. The experiment results demonstrate that RoDyn-SLAM achieves state-of-the-art performance among recent neural RGB-D methods in both accuracy and robustness. In future work, a more robust keyframe management method is a promising direction to improve the system further.

REFERENCES

[1] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot." in *IROS*, 1991.

[2] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *IROS*, 2018.

[3] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE RAL*, 2018.

[4] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *IEEE RAS*, 2019.

[5] J. Zhang, M. Henein, R. Mahony, and V. Ila, "Vdo-slam: a visual dynamic object-aware slam system," *arXiv preprint*, 2020.

[6] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable rgb-d slam in dynamic environments," *IEEE RAS*, 2018.

[7] J. Cheng, Y. Sun, and M. Q.-H. Meng, "Improving monocular visual slam in dynamic environments: An optical-flow-based approach," *Advanced Robotics*, 2019.

[8] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "Flowfusion: Dynamic dense rgb-d slam based on optical flow," in *ICRA*, 2020.

[9] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE TRO*, 2017.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE TRO*, 2021.

[11] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss, "Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals," in *IROS*, 2019.

[12] E. Sucar, S. Liu, J. Ortiz, and A. Davison, "iMAP: Implicit mapping and positioning in real-time," in *ICCV*, 2021.

[13] 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 *CVPR*, 2022.

[14] H. Wang, J. Wang, and L. Agapito, "Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam," in *CVPR*, 2023.

[15] M. M. Johari, C. Carta, and F. Fleuret, "Eslam: Efficient dense slam system based on hybrid representation of signed distance fields," in *CVPR*, 2023.

[16] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," in *ICCV*, 2021.

[17] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi, "OneFormer: One Transformer to Rule Universal Image Segmentation," in *CVPR*, 2023.

[18] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "Dynaslam ii: Tightly-coupled multi-object tracking and slam," *IEEE RAL*, 2021.

[19] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *ECCV*, 2020.

[20] 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*, 2021.

[21] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, "Di-fusion: Online implicit 3d reconstruction with deep priors," in *CVPR*, 2021.

[22] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *ISMAR*, 2022.

[23] 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 *CVPR*, 2021.

[24] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *ICCV*, 2021.

[25] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *CVPR*, 2021.

[26] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz, "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields," *ACM Trans. Graph.*, 2021.

[27] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic view synthesis from dynamic monocular video," in *ICCV*, 2021.

[28] Q.-A. Chen and A. Tsukada, "Flow supervised neural radiance fields for static-dynamic decomposition," in *ICRA*, 2022.

[29] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "Dˆ 2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video," *NeurIPS*, 2022.

[30] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, and J.-B. Huang, "Robust dynamic radiance fields," in *CVPR*, 2023.

[31] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "Robustnerf: Ignoring distractors with robust losses," in *CVPR*, 2023.

[32] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*, 2007.

[33] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *ICCV*, 2011.

[34] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, 2022.

[35] 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*, 2021.

[36] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *CVPR*, 2022.

[37] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "NeRF−−: Neural radiance fields without known camera parameters," *arXiv preprint*, 2021.

[38] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-adjusting neural radiance fields," in *ICCV*, 2021.

[39] W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-nerf: Optimising neural radiance field with no pose prior," in *CVPR*, 2023.

[40] H. Li, X. Gu, W. Yuan, L. Yang, Z. Dong, and P. Tan, "Dense rgb slam with neural implicit maps," in *ICLR*, 2023.

[41] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, 2012.

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *IROS*, 2012.

[43] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, 1987.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint*, 2014.

[45] J. Canny, "A computational approach to edge detection," *IEEE TPAMI*, 1986.

[46] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *IROS*, 2013.

[47] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," in *NeurIPS*, 2021.