RGS-SLAM: Robust Gaussian Splatting SLAM with Image Reconstruction in Dynamic Scenes

Ting-Wei Yang, Pao-Kai Wang, Hui-Wen You, Yu-Chen Lin, *Senior Member, IEEE*

***Abstract*— Simultaneous Localization and Mapping (SLAM) enables autonomous navigation and mapping for mobile robots, but most visual SLAM systems assume static environments and fail in the presence of dynamic objects, leading to localization drift and distorted maps. To address this, we propose RGS-SLAM, a robust Gaussian Splatting–based SLAM system that integrates YOLOv8 for dynamic object detection and LaMa inpainting to restore occluded regions, ensuring stable tracking and dense map construction. The system employs 3D Gaussian Splatting for efficient scene representation and photorealistic rendering. Experiments on the TUM RGB-D dynamic dataset demonstrate that RGS-SLAM reduces trajectory error by up to 91.6% compared with ORB-SLAM3 and generates high-quality dense maps under dynamic conditions. Moreover, the system runs efficiently on the NVIDIA Jetson AGX Orin, highlighting its feasibility for real-world edge applications.**

***Keywords*—** Simultaneous Localization and Mapping, Visual SLAM, Dynamic Environment, Gaussian Splatting, Dense Mapping

I. INTRODUCTION

With the rapid advancement of automation and intelligent systems, simultaneous localization and mapping (SLAM) has become a fundamental technology enabling mobile platforms to operate autonomously in complex environments. Modern SLAM solutions are generally categorized into LiDAR-based and vision-based approaches. While LiDAR SLAM benefits from accurate 3D point-cloud measurements, its higher cost and deployment constraints limit practical scalability. In contrast, visual SLAM leverages camera imagery to extract features or pixel variations, offering advantages of lower hardware cost and greater integration flexibility. As a result, vision-based SLAM has attracted increasing attention in recent years. Despite significant progress, most visual SLAM methods are built upon the assumption of static environments, treating scene objects as stationary. Although this assumption simplifies algorithm design, it does not hold in real-world scenarios populated with pedestrians, vehicles, and other moving objects. Dynamic elements introduce incorrect feature correspondences, causing unstable pose estimation, blurred map points, and ghosting artifacts. A practical visual SLAM system must therefore maintain reliable tracking under dynamic disturbances while preserving reconstruction quality. Developing a robust SLAM pipeline that operates consistently in dynamic scenes remains a core challenge for real-world deployment.

This study addresses the issues of pose drift and map degradation in visual SLAM under dynamic environments. A novel reconstruction-oriented framework, Robust Gaussian Splatting SLAM (RGS-SLAM), is proposed to handle dynamic scenes effectively. YOLOv8 [1] is incorporated to perform real-time instance segmentation and produce dynamic masks for removing motion-related features that disrupt tracking. ORB-based feature points on dynamic regions are explicitly excluded during pose estimation. To recover occluded backgrounds, the LaMa [2] image inpainting network is adopted to restore missing structure and prevent geometric artifacts. Dense mapping is then achieved using 3D Gaussian Splatting with differentiable rendering for high-fidelity reconstruction. The system was evaluated under static and dynamic conditions on three benchmark datasets: TUM RGB-D [3], BONN [4], and Replica [5]. The experimental results show that on the TUM dynamic sequences, RGS-SLAM reduced the Absolute Trajectory Error (ATE) by 91.6% compared to ORB-SLAM3 [6], surpassing existing 3DGS-based SLAM methods.

In addition, the complete RGS-SLAM pipeline is successfully executed on the NVIDIA Jetson AGX Orin embedded platform, demonstrating its feasibility for real-world dynamic environments.

II. RELATED WORK

A. Neural Implicit Visual SLAM

Traditional visual SLAM systems such as ORB-SLAM [7] and DSO achieve real-time performance through incremental pose estimation while reducing the computational cost of global optimization. However, these methods generally rely on static-scene assumptions; in dynamic environments, moving objects introduce feature mismatches and occlusions, leading to pose drift and map degradation. With the advent of deep learning, learning-based approaches such as DVO-SLAM and DROID-SLAM enhance robustness by incorporating depth and motion estimation, yet their performance can still degrade when large dynamic regions violate brightness or geometric consistency, resulting in unstable tracking. To alleviate these issues, methods such as DS-SLAM [8] and DynaSLAM [9] introduce semantic segmentation and motion masking, with the latter further incorporating background restoration. While these strategies effectively suppress dynamic regions, they may over-mask areas near motion boundaries, discarding valid static features and reducing feature support for pose estimation, which can negatively impact localization stability in real-world dynamic scenes.

*Research supported by National Science and Technology Council, Taiwan, R.O.C., under the contract No. NSTC 112-2628-E-035-001-MY3. and No. NSTC 114-2218-E-035-001 (Corresponding author: Yu-Chen Lin; e-mail: yuchlin@fcu.edu.tw)

T. W. Yang, P. K. Wang, H. W. You, and Y. C. Lin are with the Department of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, R.O.C. (e-mail: tingweiyang713@gmail.com; zxcvbnm28196544@gmail.com; 9576mq260@gmail.com; yuchlin@fcu.edu.tw)

B. Gaussian Splatting SLAM

3D Gaussian Splatting (3DGS) [10] provides an explicit and differentiable scene representation that supports high-quality rendering and real-time performance. Unlike NeRF [11], which uses multilayer perceptrons for implicit radiance-field modeling, 3DGS represents a scene with discrete 3D Gaussian ellipsoids. Each Gaussian is parameterized by its center position, orientation, scale, color, and opacity, forming a compact yet expressive representation of scene geometry and appearance. The framework initializes the scene from a sparse Structure-from-Motion (SfM) point cloud and uses splatting-based projection, where a tile-based rasterizer sorts and blends Gaussians to render pixel values on the image plane. By jointly optimizing position, scale, orientation, and appearance parameters under photometric supervision, 3DGS achieves high rendering fidelity and efficient convergence, making it suitable for online SLAM applications.

In addition, several studies have integrated 3DGS into SLAM frameworks. MonoGS [12] first demonstrated the feasibility of using Gaussians as the sole 3D map representation in monocular SLAM, integrating tracking, mapping, and optimization while introducing geometric regularization to reduce reconstruction ambiguity. Gaussian-SLAM [13] improved scalability through submap division, enabling independent optimization of spatial regions for large-scale scenes and dense real-time reconstruction without sacrificing accuracy. These works show that 3DGS-based approaches can outperform NeRF-style SLAM in efficiency and visual quality in static environments. However, they generally assume static scenes and thus struggle in real-world dynamic environments, where moving objects cause inconsistent observations and appearance blending artifacts. Recent work such as DGS-SLAM [14] extends 3DGS to dynamic settings using dynamic-object removal and motion segmentation. Nevertheless, challenges remain in handling background occlusions, maintaining temporal consistency during reconstruction, and preventing performance degradation when dynamic objects dominate the field of view.

To address these limitations, this study proposes Robust Gaussian Splatting Visual SLAM (RGS-SLAM), which integrates dynamic-object detection, mask-guided feature filtering, and image inpainting to restore the occluded static regions. This design improves both map completeness and localization accuracy, enabling robust and visually consistent dense reconstruction in complex dynamic environments.

III. METHODOLOGY

The architecture of the proposed Robust Gaussian Splatting SLAM (RGS-SLAM) system is illustrated in Fig. 1. The system consists of four main modules: dynamic object detection, tracking and localization, image restoration, and mapping. RGB-D images are provided to both detection and tracking modules. The detection module employs YOLOv8 [1] for instance segmentation to generate masks of dynamic objects. These masks are then forwarded to the tracking and image restoration modules to facilitate robust handling of dynamic regions during pose estimation and dense reconstruction.

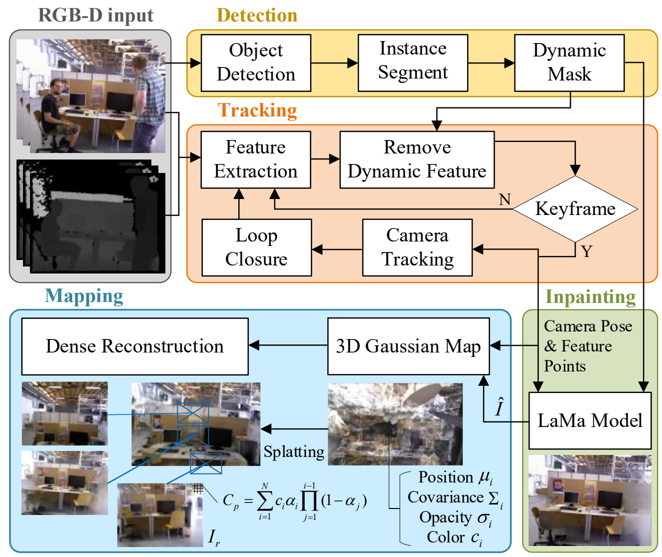
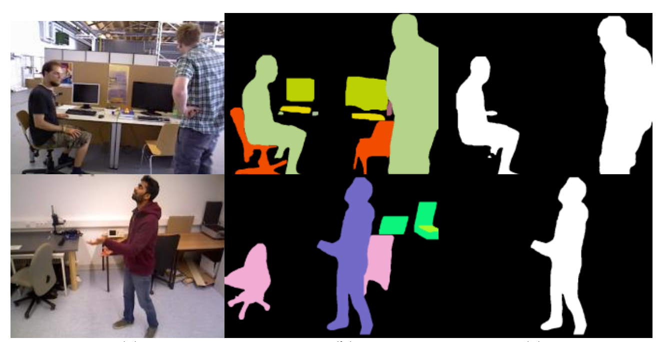


Figure 1. **SLAM System Architecture:** Our system integrates dynamic perception, robust tracking, and 3D Gaussian mapping, unifying detection, tracking, and reconstruction to achieve real-time, high-density 3D SLAM.

A. Detection

To obtain accurate and real-time dynamic-region masks, this study adopts YOLOv8s-Seg [1], pretrained on the COCO dataset, for instance segmentation. YOLOv8s-Seg provides a favorable balance between segmentation accuracy and inference speed, making it suitable for dynamic-object perception in SLAM. Compared with bounding-box detection, instance segmentation provides pixel-level masks that better preserve object contours. For RGB images from TUM RGB-D [3] and BONN [4], YOLOv8s-Seg generates instance masks, from which masks of predefined dynamic object classes are selected and converted into a binary dynamic mask. Dynamic regions are determined by object categories rather than actual motion states; for example, if cars are defined as dynamic objects, all car instances are marked as dynamic regions. As shown in Fig. 2, the mask is used for dynamic-region removal and image restoration to improve tracking stability and SLAM accuracy in dynamic environments.



(a) (b) (c)

Figure 2. Binary dynamic mask generation using YOLOv8: (a) input RGB image, (b) instance segmentation results, and (c) binary dynamic mask obtained by selecting predefined dynamic object classes.

B. Camera-Tracking

For camera tracking, RGS-SLAM uses ORB-SLAM3 [6] as the tracking backbone and combines it with dynamic object masks generated by YOLOv8 to improve localization robustness in dynamic environments. ORB features are extracted using FAST corners and rotated BRIEF descriptors, together with an image pyramid for scale and rotation invariance. ORB-SLAM3 further supports stable feature matching across frames through multi-scale keypoint extraction and intensity-centroid-based orientation estimation. To reduce motion-induced drift, keypoints inside the binary dynamic mask are removed before feature matching, preventing features on moving objects from affecting pose optimization. By keeping only static and geometrically reliable points, the system provides cleaner constraints for bundle adjustment and achieves more stable frame-to-frame tracking under dynamic motion.

Keyframes are inserted based on the stability of tracked features. Let N_t , t , N_k denote the number of tracked features in the current frame, the number of features in the latest keyframe, and the number of matched features between them, respectively. A new keyframe is inserted when:

$$N_t \geq N_{\min} \wedge (\frac{M_{t,k}}{N_k} \leq \rho \vee (t\text{-}k) \geq T_{gap}) \tag{1}$$

where $N_{\min}$ is the minimum required number of tracking features, and T_{gap} is the maximum keyframe interval. In this study, $N_{\min}$=50 , $\rho = 0.9$, and T_{gap}= 20 are empirically set based on the ORB-SLAM3 keyframe insertion strategy to balance tracking stability and keyframe redundancy. After feature matching, camera motion is estimated from matched ORB descriptors, and loop closure is used to reduce accumulated drift.

C. Dynamic-Region Inpainting

Dynamic-object removal in SLAM often creates missing image regions, causing texture loss and incomplete scene information that may affect feature extraction and mapping accuracy. To preserve scene completeness, RGS-SLAM uses a LaMa-based [2] image restoration module to fill masked dynamic regions and recover occluded background structures. The restored images provide more consistent visual inputs for later tracking and mapping, instead of relying only on the detection mask. Given an RGB image $I \in \mathbb{R}^{H \times W \times 3}$ and a binary dynamic-object mask $m \in \{0,1\}^{H \times W}$, the masked image is formed as $I_m = I \odot m$, where the operator $\odot$ denotes pixel-wise multiplication between the image and the mask. The concatenated input to the inpainting network is

$$X = stack(I_m, m) \in \mathbb{R}^{H \times W \times 4} \tag{2}$$

and the restored image is predicted using a pretrained LaMa generator $f_\theta(\cdot)$:

$$\hat{I} = f_\theta(\mathrm{X}) \in \mathbb{R}^{H \times W \times 3} \tag{3}$$

The restored image, denoted as $\hat{I}$, supplies complete background textures to the SLAM pipeline and provides more stable visual observations under dynamic conditions. By recovering regions occluded by moving objects, $\hat{I}$ helps maintain appearance consistency across frames and provides ORB-SLAM3 [6] with improved inputs for pose estimation.

D. Mapping

In the mapping stage, RGS-SLAM adopts 3DGS [10] as the unified scene representation. Based on the camera poses and feature correspondences estimated by ORB-SLAM3 [6], each tracked feature is initialized as an anisotropic 3D Gaussian. The index *i* denotes the *i*-th Gaussian in the map. Each Gaussian is represented by its center, covariance matrix, SH-based color coefficients, and opacity. Its spatial density is defined as:

$$G(x) = e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \tag{4}$$

where x is a 3D point, μ_i is the center of the i-th Gaussian, and Σ_i is its covariance matrix, which controls anisotropic scale and orientation. Given the camera pose, the Gaussian center is projected onto the image plane as:

$$\mu_I = \pi(T_w^c, \mu_w) \tag{5}$$

where T_w^c is the world-to-camera transform, μ_w is the Gaussian center in the world coordinate frame, μ_I is the projected 2D image position, and π is the camera projection function. To avoid under- or over-reconstruction, adaptive density control is applied through clone and split operations based on opacity and spatial coverage. The map is optimized by minimizing the photometric loss between the rendered image and the restored target image:

$$L_p = (1-\lambda)\cdot \left\| I_r - I_{gt} \right\|_1 + \lambda \cdot (1 - SSIM(I_r, I_{gt})) \tag{6}$$

where I_r is the image rendered from the current 3D Gaussian map, and I_{gt} is the restored target image generated by the LaMa inpainting module in Section III.C. Therefore, I_{gt} corresponds to the restored image $\hat{I}$ in Section III.C, while I denotes the original RGB input before masking and restoration. The SSIM term measures structural similarity between I_r and I_{gt} , and $(1 - SSIM(I_r, I_{gt}))$ is used to penalize structural inconsistency. The L1 term preserves pixel-level color accuracy, and λ balances these two terms. With SLAM-based Gaussian initialization, dynamic-region-free supervision, and adaptive density refinement, RGS-SLAM incrementally builds a dense and geometrically consistent 3D Gaussian map in dynamic environments.

IV. EXPERIMENTAL RESULTS

RGS-SLAM is evaluated in both dynamic and static environments, with a focus on tracking accuracy, reconstruction quality, dynamic-region inpainting, and computational performance.

A. Implementation and Datasets

RGS-SLAM integrates ORB-SLAM3 [6] for tracking, 3D Gaussian Splatting [10] for dense mapping, and LaMa inpainting [2] for dynamic-region restoration. The system is implemented on a desktop platform with an i7-10700 CPU, 32 GB RAM, and an NVIDIA GeForce RTX 3090 GPU. The full

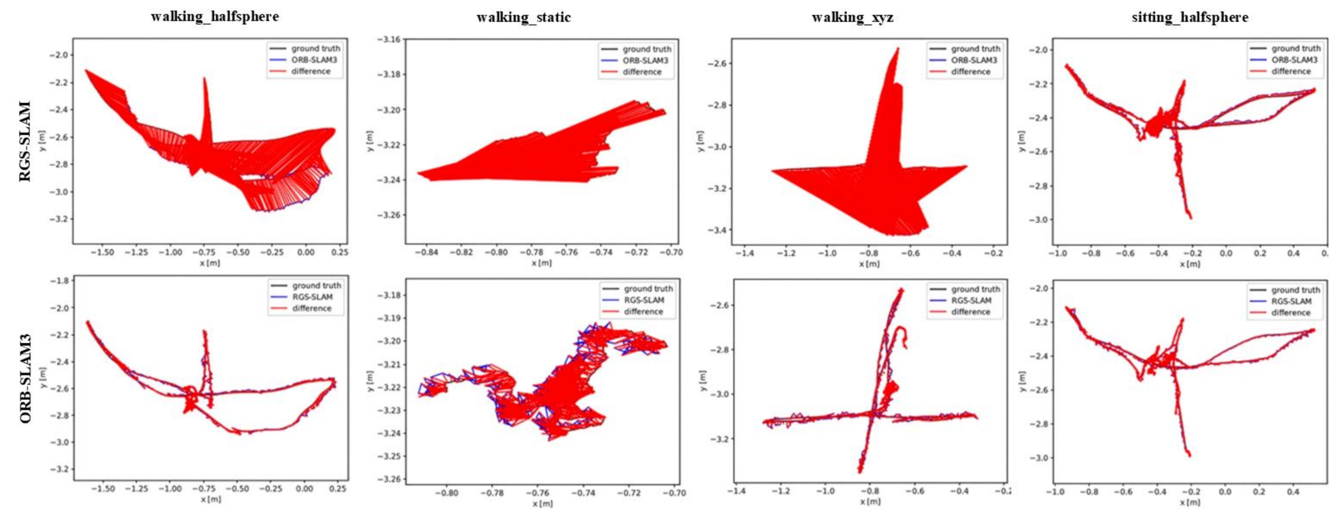


Figure 3. Comparison of camera trajectories estimated by ORB-SLAM3 and RGS-SLAM with TUM RGB-D dataset.

pipeline is also deployed on an NVIDIA Jetson AGX Orin, showing that detection, masking, inpainting, tracking, and Gaussian-splatting reconstruction can run within a unified embedded workflow.

The system is evaluated on three public datasets: TUM RGB-D [3], BONN [4], and Replica [5]. TUM RGB-D and BONN contain dynamic indoor scenes with people and handheld objects that often occlude the camera view, making them suitable for evaluating robustness under dynamic motion. Replica provides high-quality static indoor scenes for assessing geometric and photometric reconstruction quality. Tracking accuracy is evaluated using Absolute Trajectory Error (ATE), reported as RMSE and standard deviation (SD), while reconstruction quality is measured by PSNR and SSIM between rendered and ground-truth images.

B. Trajectory Accuracy and Evaluation

To comprehensively evaluate the tracking performance of the proposed SLAM system, comparative analyses are conducted in high-dynamic, low-dynamic, and static environments. Fig. 3 and Table I report the results on dynamic scenes from the TUM RGB-D dataset, where moving objects frequently occlude the camera view and introduce challenging conditions for pose estimation.

For dynamic scenes from the TUM RGB-D dataset [3], RGS-SLAM achieves significantly lower ATE than ORB-SLAM3 [6] and Gaussian-SLAM [13], particularly in highly dynamic sequences such as walking_xyz and walking_halfsphere, as summarized in Table I. The estimated trajectories remain close to the ground truth, and the standard deviation of ATE is consistently smaller than that of feature-based and NeRF-based baselines, indicating stable tracking under strong motion and occlusions. The BONN dynamic dataset in Table II contains complex human motions and object interactions, where several classical and NeRF-based methods fail to maintain stable tracking in some sequences. In contrast, RGS-SLAM achieves stable tracking across all evaluated dynamic sequences, indicating that combining dynamic object masking with inpainted supervision effectively mitigates the impact of moving objects on pose estimation.

TABLE I. Performance on TUM Dynamic Scenes (ATE: cm)[6, 12-14]

Sequences	**Method**	**ORB-SLAM3**	**NICE-SLAM**	**Co-SLAM**	**Mono GS**	**GS-SLAM**	**DGS-SLAM**	**RGS-SLAM (ours jetson)**	**RGS-SLAM (ours)**
	Year	2021	2022	2023	2024	2024	2024		
fr3/walking_xyz	RMSE	28.1	113.8	51.8	73.4	37.2	4.1	**1.2**	**1.2**
	SD	12.2	42.9	25.3	20.1	9.9	2.2	**0.6**	**0.6**
fr3/walking_halfsphere	RMSE	30.5	X	105.1	65.6	60.0	5.5	2.2	**1.6**
	SD	9.0	X	42.0	24.8	20.7	2.8	1.3	**0.8**
fr3/walking_static	RMSE	2.0	88.2	49.5	5.5	8.4	**0.6**	**0.6**	0.9
	SD	1.1	27.8	10.8	3.0	4.1	**0.2**	**0.2**	0.4
fr3/sitting_halfsphere	RMSE	2.6	45.0	4.7	2.7	7.4	4.1	1.3	1.5
	SD	1.6	14.4	2.2	1.5	5.4	1.6	0.6	**0.6**
Avg.	RMSE	15.8	82.3	52.3	36.8	28.3	3.6	1.3	**1.3**
	SD	6.0	28.4	20.0	12.4	10.0	1.7	0.7	**0.6**

TABLE II. Performance on BONN Dynamic Scenes (ATE: cm)

Sequences	**Method**	**ORB-SLAM3**	**NICE-SLAM**	**Co-SLAM**	**Mono GS**	**GS-SLAM**	**DGS-SLAM**	**RGS-SLAM (ours jetson)**	**RGS-SLAM (ours)**
	Year	2021	2022	2023	2024	2024	2024		
balloon	RMSE	5.8	X	28.8	33.2	39.5	2.9	**2.6**	2.8
	SD	2.8	X	9.6	16.4	19.3	**0.8**	1.4	1.4
balloon2	RMSE	17.7	66.8	20.6	26.5	35.6	6.0	**2.4**	2.6
	SD	8.6	20.0	8.1	14.2	19.5	2.8	**1.0**	**1.0**
ps_track	RMSE	70.7	54.9	61.0	63.2	93.3	9.8	3.4	**3.3**
	SD	32.6	27.5	22.2	29.0	36.3	4.1	**1.3**	1.4
ps_track2	RMSE	77.9	45.3	59.1	47.2	51.2	11.1	6.9	5.5
	SD	43.8	17.5	24.0	15.4	19.1	3.9	3.1	3.3
ball_track	RMSE	**3.1**	21.2	38.3	4.3	-	5.6	3.8	3.5
	SD	**1.6**	13.1	17.4	2.2	-	2.8	1.9	1.8
mv_box2	RMSE	3.5	31.9	70.0	22.9	6.1	8.8	3.2	**3.1**
	SD	1.5	13.6	25.5	12.4	4.5	3.8	**1.3**	1.5
Avg.	RMSE	29.8	44.0	46.3	32.9	45.1	7.3	3.7	3.5
	SD	15.2	18.3	17.8	14.9	19.7	3.0	1.7	1.7

For static TUM RGB-D sequences in Table III, RGS-SLAM attains trajectory accuracy comparable to other 3DGS-based SLAM methods. The additional dynamic-object handling modules do not degrade localization in static environments, indicating that the system maintains precise tracking even when no large dynamic objects are present.

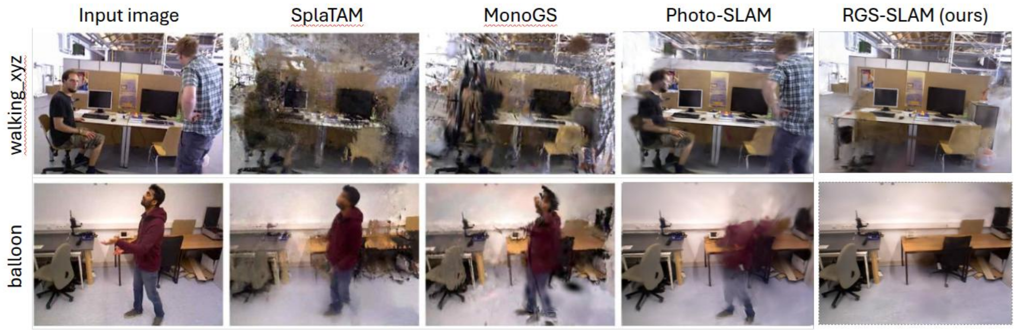


Figure 4. Qualitative comparison of rendered results from different SLAM methods on dynamic scenes in the TUM RGB-D [3] dataset.

Figure 5. Additional qualitative results of **RGS-SLAM (Ours)** on dynamic sequences from the TUM RGB-D [3] dataset.

TABLE III. Performance on TUM RGB-D Static Scenes (ATE [cm])

Method	ORB-SLAM3	NICE-SLAM	Co-SLAM	MonoGS	GS-SLAM	DGS-SLAM	RGS-SLAM (ours jetson)	RGS-SLAM (ours)
	2021	2022	2023	2024	2024			
fr1/desk	1.7	2.7	2.4	**1.5**	3.3	1.6	1.9	1.7
fr2/xyz	0.4	1.8	1.7	1.4	1.3	0.3	**0.2**	0.4
fr3/office	1.7	3	2.4	1.5	6.6	**0.7**	0.9	1.7

C. Reconstruction and Rendering Quality

To evaluate mapping performance, both qualitative and quantitative reconstruction results are analyzed on dynamic scenes from the TUM RGB-D [3] and Replica [5] datasets, where clean static backgrounds are unavailable. As shown in Fig. 4, existing SLAM methods often fuse moving objects into the reconstructed scene, resulting in ghosting artifacts and degraded geometry. By removing dynamic regions and restoring occluded backgrounds via image inpainting, RGS-SLAM produces cleaner and more coherent reconstructions under dynamic conditions. Additional qualitative results of RGS-SLAM on the TUM RGB-D dataset are presented in Fig. 5, demonstrating stable reconstruction performance under varying motion patterns and occlusions. This design improves reconstruction continuity by providing more reliable background observations for subsequent mapping and rendering optimization.

Quantitative reconstruction quality is reported in Tables IV and V using PSNR and SSIM, which are widely adopted to evaluate dense reconstruction and rendering fidelity in SLAM and neural rendering systems [10, 12, 13]. As shown in the results, RGS-SLAM achieves competitive reconstruction quality compared with recent Gaussian-based SLAM methods such as MonoGS and GS-SLAM, rather than consistently surpassing them in every sequence. This indicates that the proposed dynamic-region removal and image inpainting modules may introduce slight local appearance differences during reconstruction. Nevertheless, by explicitly handling moving objects and restoring the background regions occluded, RGS-SLAM maintains stable reconstruction quality while reducing the negative effects of dynamic regions on mapping. Overall, the results suggest that the proposed framework provides a practical balance between dynamic-scene robustness and reconstruction fidelity, making it suitable for long-term operation in real-world dynamic environments.

TABLE IV. Reconstruction Quality on TUM (PSNR / SSIM)

Method		NICE-SLAM	Co-SLAM	MonoGS	RGS-SLAM (ours jetson)	RGS-SLAM (oues)
	Year	2022	2023	2024		
fr1_desk	PSNR	12.00	16.42	18.67	21.67	**22.62**
	SSIM	0.42	0.48	0.71	0.77	**0.8**
fr2_xyz	PSNR	18.20	19.17	15.94	**26.2**	25.03
	SSIM	0.60	0.59	0.80	**0.85**	0.81
fr3_office	PSNR	16.34	17.86	19.23	24.42	**25.8**
	SSIM	0.55	0.54	0.74	0.82	**0.86**

TABLE V. Reconstruction Quality on Replica (PSNR / SSIM)

Method		NICE-SLAM	Co-SLAM	MonoGS	GS-SLAM	RGS-SLAM (ours jetson)	RGS-SLAM (ours)
	Year	2022	2023	2024	2024	2022	
room0	PSNR	22.12	27.27	**34.83**	31.56	31.90	31.87
	SSIM	0.689	0.910	0.954	**0.968**	0.920	0.918
room1	PSNR	22.47	28.45	**36.43**	32.86	36.23	34.22
	SSIM	0.757	0.909	0.959	**0.973**	0.956	0.942
room2	PSNR	24.52	29.06	**37.49**	32.59	35.69	37.26
	SSIM	0.814	0.932	0.965	0.971	0.962	**0.972**
office0	PSNR	29.07	34.14	39.95	38.70	**40.05**	38.61
	SSIM	0.886	0.961	0.971	**0.986**	0.974	0.97
office1	PSNR	30.34	34.87	**42.09**	41.17	40.82	41.49
	SSIM	0.886	0.969	0.977	**0.993**	0.969	0.973
office2	PSNR	19.66	28.43	**36.24**	32.36	34.08	34.10
	SSIM	0.797	0.938	0.964	**0.978**	0.954	0.954
office3	PSNR	22.23	28.76	**36.70**	32.03	34.15	34.44
	SSIM	0.801	0.941	0.963	**0.970**	0.947	0.95
office4	PSNR	24.94	30.91	36.07	32.92	36.81	37.50
	SSIM	0.856	0.955	0.960	0.968	0.960	0.967

D. Effect of Dynamic-Region Inpainting

The effect of the dynamic-region inpainting module is evaluated by comparing three configurations. Without dynamic masking or inpainting, moving objects are directly fused into the reconstruction, leading to severe ghosting artifacts and accumulated pose drift, as commonly observed in dense SLAM systems under dynamic scenes [10]. Enabling dynamic masks alone prevents dynamic structures from being integrated into the map and improves tracking stability; however, it also introduces large missing regions in the rendered images, resulting in discontinuous geometry and incomplete supervision for 3D Gaussian Splatting optimization. When LaMa-based inpainting [2] is applied on top of the dynamic masks, the missing background content is restored with spatially plausible textures, providing more complete and temporally consistent inputs for dense mapping. As illustrated in Fig. 6, inpainting effectively reduces background discontinuities and artifacts around occluded regions, yielding cleaner reconstructions and more coherent geometry. Overall, these results indicate that dynamic-region inpainting not only enhances visual completeness but also contributes to stable and reliable mapping in dynamic environments, benefiting downstream pose estimation and mapping in ORB-SLAM3-based tracking pipelines [6].

Figure 6. Dynamic-region inpainting results. Left: binary dynamic mask. Right: inpainted result after restoring occluded background regions.

E. Ablation Study

To evaluate the effectiveness of the dynamic object removal and image inpainting modules in RGS-SLAM, an ablation study is conducted on the TUM-RGB-D dataset using the ATE as the evaluation metric. As shown in Table VI, large localization errors are observed in highly dynamic scenes when dynamic objects are not removed. Enabling dynamic object removal significantly reduces the error, while further incorporating image inpainting provides additional improvements in most dynamic sequences, demonstrating the complementary effects of the two modules. In low-dynamic scenes, the performance differences among configurations are marginal, indicating that the proposed method does not degrade SLAM performance in static environments.

TABLE VI. Ablation on removal and inpainting (RMS ATE, cm).

Sequence	w/o Remove	w/o Inpainting	Ours
fr3/walking_xyz	26.678	1.241	**1.208**
fr3/walking_halfsphere	34.595	1.916	**1.625**
fr3/walking_static	1.595	**0.586**	0.876
fr3/sitting_halfsphere	0.831	0.925	**0.912**

V. Conclusion

This paper presents RGS-SLAM, a robust Gaussian splatting-based visual SLAM system designed for dynamic environments. By integrating dynamic-object detection, image inpainting, and dense 3D Gaussian mapping, the proposed framework mitigates localization drift and map degradation caused by dynamic motion. Experimental evaluations on the TUM RGB-D, BONN, and Replica datasets demonstrate that RGS-SLAM achieves improved localization accuracy and more complete reconstructions in dynamic scenes compared with existing methods. In particular, the proposed dynamic-region handling strategy enables stable tracking and consistent dense mapping under strong motion and occlusion. Furthermore, the deployment of the pipeline on an NVIDIA Jetson AGX Orin confirms its feasibility for embedded and real-world applications. Future work will focus on improving dynamic mask accuracy, accelerating the inpainting process, and incorporating semantic information to further enhance robustness, efficiency, and scene understanding.

References

[1] G. Jocher, *et al.*, “Ultralytics YOLOv8,” [Online] Available: https://github.com/ultralytics/ultralytics, 2023.

[2] R. Suvorov *et al.*, “Resolution-robust large mask inpainting with Fourier convolutions,” *IEEE/CVF Winter Conf. Applications of Computer Vision (WACV)*, pp. 2149–2159, 2022.

[3] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A Benchmark for the Evaluation of RGB-D SLAM Systems,” *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 573-580, 2012.

[4] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss, “ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals,” *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 7855-7862, 2019.

[5] J. Straub, T. Whelan, and L. Ma et al. “The Replica Dataset: A Digital Replica of Indoor Spaces,” *arXiv: 1906.05797*, pp. 1-10, 2019.

[6] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM,” *IEEE Trans. Robotics*, vol. 37, no. 6, pp. 1874-1890, 2021.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Trans. Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.

[8] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, and Q. Wei, “DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments,” *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1168-1174, 2018.

[9] B. Bescos, J. M. Facil, J. Civera and J. Neira, “DynaSLAM: Tracking Mapping and Inpainting in Dynamic Scenes,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076-4083, 2018.

[10] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, “3D Gaussian Splatting for Real-Time Radiance Field Rendering,” *ACM Trans. Graphics*, vol. 42, no. 4, pp. 1-14, 2023.

[11] 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.

[12] H. Matsuki, R. Murai, P. H. Kelly and A. J. Davison, “Gaussian Splatting SLAM,” *IEEE/CVF Conf. Computer Vision and Pattern Recognition*, pp. 18039-18048, 2024.

[13] V. Yugay, Y. Li, T. Gevers, and M. R. Oswald, “Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting,” *arXiv: 2312.10070*, pp. 1-24, 2023.

[14] M. Kong, J. Lee, S. Lee, and E. Kim, “DGS-SLAM: Gaussian Splatting SLAM in Dynamic Environment,” *arXiv: 2411.10722*, pp. 1-7, 2024.