Scene Reconstruction with a Non-Rigid Environment - Dynamic SLAM Techniques

Katerina Nikiforova
knikifor@andrew.cmu.edu

Morgan Mayborne
mmayborn@andrew.cmu.edu

Sai Deepa Vaddi
svaddi@andrew.cmu.edu

Abstract

Dynamic environments pose significant challenges for traditional Simultaneous Localization and Mapping (SLAM) systems, which typically struggle with moving objects and scene complexity. This research introduces an innovative approach to enhance ORB-SLAM2 by integrating advanced computer vision techniques—specifically Mask R-CNN and SAM 2—to effectively detect, segment, and filter dynamic objects during scene reconstruction. By systematically identifying and removing moving elements from the mapping pipeline, our method significantly improves localization accuracy and robustness in non-rigid environments. Experimental results demonstrate improvements compared to baseline ORB-SLAM2, highlighting the potential of deep learning-enhanced SLAM techniques for real-world robotic navigation and perception systems.

1 Introduction

Simultaneous Localization and Mapping (SLAM) represents a critical technological frontier in autonomous systems, enabling robots, drones, and intelligent agents to navigate and comprehend complex environments. Traditional SLAM algorithms, exemplified by ORB-SLAM2, have made remarkable strides in creating consistent spatial representations. However, these systems fundamentally assume a static world, rendering them vulnerable when confronted with dynamic, ever-changing environments.

Real-world scenarios are inherently complex and dynamic. Streets, indoor spaces, and interactive environments are characterized by constant motion—pedestrians walking, vehicles moving, objects being manipulated. These dynamics introduce significant noise and uncertainty into traditional SLAM pipelines. Existing algorithms often struggle to differentiate between static scene geometry and transient, moving elements, leading to inaccurate camera pose estimation, inconsistent map representations, reduced localization reliability, and increased computational complexity.

Our research is motivated by the critical need to develop SLAM systems that can seamlessly adapt to real-world complexity. By integrating state-of-the-art computer vision techniques, we aim to transform SLAM from a rigid, static-world approach to a flexible, intelligent mapping paradigm. The proposed approach introduces a novel three-stage dynamic object handling mechanism: 1) Object Detection: Utilizing Mask R-CNN to precisely identify and segment dynamic objects within each keyframe; 2) Dynamic Region Analysis: Employing SAM 2 to predict and track keypoints associated with moving elements; and 3) Selective Filtering: Systematically removing detected dynamic points from the ORB-SLAM2 processing pipeline. This methodology allows for a more nuanced understanding of scene dynamics, enabling more accurate and robust scene reconstruction.

Beyond immediate robotics applications, our research contributes to several domains including autonomous navigation, augmented and virtual reality, intelligent surveillance systems, robotic perception, and urban mapping and modeling. Our specific research objectives include developing a robust dynamic object detection mechanism, enhancing ORB-SLAM2's performance in non-rigid environments, creating a generalizable framework for dynamic scene understanding, and demonstrating improved localization accuracy.

By intelligently integrating deep learning-based object detection and segmentation techniques into the traditional SLAM framework, we can significantly improve scene reconstruction capabilities in dynamic environments, transforming how autonomous systems perceive and navigate complex spaces. The subsequent sections will detail our methodology, experimental setup, results, and implications, presenting a comprehensive exploration of dynamic SLAM enhancement techniques.

2 Related Work

Visual SLAM techniques have evolved significantly over the years, with feature-based methods like ORB-SLAM2 becoming widely adopted due to their efficiency and accuracy in static environments. However, these algorithms typically assume a static world, which limits their applicability in dynamic scenes (Mur-Artal and Tardós, 2017). Recent research has focused on developing SLAM systems capable of operating in dynamic environments. Wang et al. proposed a real-time dynamic SLAM system that runs solely on CPU by incorporating a mask prediction mechanism (Wang et al., 2022). This approach demonstrates the feasibility of integrating deep learning methods into SLAM without relying on GPU support.

The integration of object detection and segmentation techniques into SLAM has gained traction as a means to improve robustness in dynamic environments. Mask R-CNN, developed by He et al., has emerged as a powerful tool for instance segmentation, enabling precise identification and outlining of objects in images (He et al., 2017). This capability makes it particularly suitable for detecting dynamic elements in SLAM scenarios. Several researchers have leveraged Mask R-CNN for dynamic SLAM applications. Yu et al. proposed DS-SLAM, which combines SegNet with optical flow tracking to eliminate dynamic feature points and improve pose estimation accuracy (Yu et al., 2018). Bescos et al. introduced DynaSLAM, which merges multiview geometry with Mask R-CNN for dynamic object detection and background reconstruction (Bescos et al., 2018).

Semantic SLAM systems aim to bridge the gap between low-level feature representations and high-level semantic understanding of the environment. Yuan et al. proposed the SaD-SLAM system, which combines semantic and depth information to extract feature points of dynamic objects and detect their states (Yuan et al., 2022). This approach improves both map and camera pose estimation by leveraging semantic information for dynamic object handling. Addressing the challenge of moving objects, some researchers have proposed integrating multi-target tracking with SLAM. Fang et al. developed a deep learning approach for moving object tracking using ML-RANSAC within an Extended Kalman Filter framework, demonstrating the ability to maintain feature tracking even with intermittent observations (Fang et al., 2019).

Our approach builds upon these advancements by combining the strengths of ORB-SLAM2, Mask R-CNN, and SAM 2. By integrating object detection, segmentation, and keypoint prediction, we aim to create a SLAM system that is inherently robust to dynamic objects, thus improving scene reconstruction in non-rigid environments.

3 Methodology

Our approach integrates three powerful computer vision techniques to enhance the robustness of SLAM in dynamic environments: ORB-SLAM2, Mask R-CNN, and SAM 2. This section details the architecture and functionality of each component, as well as their integration into our proposed system.

3.1 ORB-SLAM2

ORB-SLAM2 is a state-of-the-art, feature-based Simultaneous Localization and Mapping (SLAM) system that operates in real-time on standard CPUs for monocular, stereo, and RGB-D cameras. As depicted in Figure 1, the system's architecture is built upon three main components that run in parallel threads:

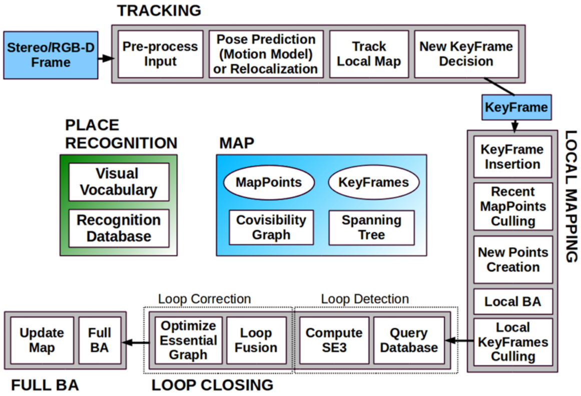

Figure 1: ORB-SLAM2 Architecture

1. **Tracking Thread**: This thread is responsible for localizing the camera with every new frame and deciding when to insert a new keyframe. It first performs an initial pose estimation by using a constant velocity model or, if that fails, by relocalizing the camera using a bag of words approach. Then, the thread tracks local map points, optimizing the

pose using motion-only bundle adjustment. If tracking is successful, it searches for additional matches with local map points and further refines the pose. Finally, it decides whether to insert a new keyframe based on specific criteria such as the number of tracked points and time since the last keyframe.

2. **Local Mapping Thread**: This thread processes new keyframes and performs local bundle adjustment to achieve optimal reconstruction in the surroundings of the camera pose. It first inserts the new keyframe, updating the covisibility graph and spanning tree, essential data structures for the system. Then, it culls redundant points to maintain a compact and efficient map. The thread also creates new map points by triangulating ORB features matched between the new keyframe and its connected keyframes. Finally, it performs local bundle adjustment to optimize the current keyframe, its neighbors, and the points they observe.

3. **Loop Closing Thread**: This thread searches for loops every time a new keyframe is inserted. If a loop is detected, it computes a similarity transformation that informs about the drift accumulated in the loop. Then it optimizes the essential graph (a sparse graph of keyframes with edges connecting covisible keyframes and loop closure keyframes) to achieve global consistency. Finally, it launches a full bundle adjustment in a separate thread to achieve optimal accuracy.

ORB-SLAM2's robust performance in various environments and its ability to reuse the map for relocation make it an ideal base for our dynamic SLAM system. However, its assumption of a static environment presents significant challenges when dealing with dynamic scenes:

- **Feature Mismatch:** ORB-SLAM2 relies on matching features between frames to estimate camera motion. In dynamic environments, features on moving objects can lead to incorrect matches, resulting in erroneous pose estimation(Wang et al., 2022).

- **Map Corruption:** Moving objects can introduce inconsistencies in the map, as features from these objects may be incorrectly incorporated as static landmarks(Wang et al., 2022).

- **Tracking Failures:** Rapid changes in the scene due to dynamic objects can cause the system to lose track of its position, especially if a large portion of the frame is occupied by moving elements(Wang et al., 2022).

- **Loop Closure Errors:** Dynamic objects can interfere with loop closure detection, leading to incorrect loop identifications or failures to detect valid loops.

- **Reduced Accuracy:** The presence of dynamic objects can significantly degrade the overall accuracy of pose estimation and mapping.

These limitations of ORB-SLAM2 in dynamic environments necessitate the integration of advanced object detection and segmentation techniques, such as Mask R-CNN and SAM 2, to effectively handle moving objects and improve SLAM performance in real-world scenarios.

3.2 Mask R-CNN

Mask R-CNN, illustrated in Figure 2, is an extension of Faster R-CNN that adds a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression. The architecture consists of several key components:

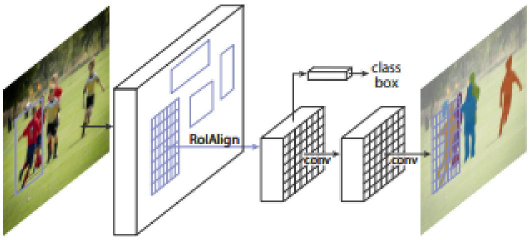

Figure 2: Mask R-CNN Architecture

1. **Backbone Network**: This is typically a convolutional network (e.g., ResNet50 or ResNet101) pre-trained on a large dataset like ImageNet. It serves as a feature extractor, producing a feature map of the entire input image.

2. **Region Proposal Network (RPN)**: The RPN scans the feature map with a sliding window and proposes regions that potentially contain objects. It outputs a set of rectangular object proposals, each with an objectness score.

3. **RoI Align**: This layer performs a precise spatial extraction of features from the feature map for each proposed region. Unlike its predecessor (RoI Pooling), RoI Align uses bilinear interpolation to compute the exact values of the input features at four regularly sampled locations in each RoI bin, preserving spatial information.

4. **Network Head**: This consists of two parallel branches:

    - The first branch performs classification and bounding-box regression.
    - The second branch generates a binary mask for each RoI, predicting whether each pixel belongs to the object or not.

Mask R-CNN's ability to provide accurate, pixel-level segmentation masks for multiple object instances makes it invaluable for identifying dynamic objects in our SLAM system. By leveraging these masks, we can effectively isolate and exclude features associated with moving objects, thereby enhancing the robustness of our mapping and localization processes.

3.3 SAM 2

SAM 2 (Segment Anything Model 2) is an advanced, promptable segmentation model designed specifically for video applications. Building upon the success of its predecessor (SAM), SAM 2 introduces several key innovations that make it particularly suitable for our dynamic SLAM system, as shown in Figure 3:

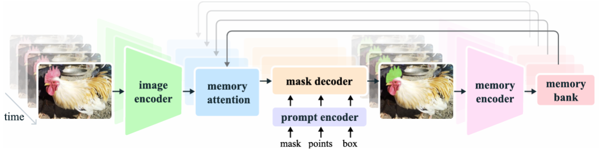

Figure 3: SAM 2 Architecture

1. **Per-session Memory Module**: This module maintains information about target objects throughout a video sequence. It allows the model to consistently track objects even when they temporarily leave the frame or become occluded, a common occurrence in real-world SLAM scenarios.

2. **Streaming Architecture**: Unlike traditional segmentation models that process entire videos at once, SAM 2 is designed to process video frames sequentially. This architecture aligns perfectly with the real-time nature of SLAM systems, allowing for frame-by-frame processing and immediate integration of segmentation results.

3. **Correction Capability**: SAM 2 allows for on-the-fly adjustments to mask predictions based on additional prompts. This feature is particularly useful in SLAM applications where initial segmentations may need refinement based on subsequent frames or additional sensor data.

4. **Multi-object Tracking**: The model can simultaneously track and segment multiple objects in a video, which is crucial for handling complex, dynamic environments with multiple moving entities.

5. **Temporal Consistency**: SAM 2 maintains consistent segmentations across frames, reducing jitter and improving the stability of object tracking over time.

By incorporating SAM 2 into our system, we can accurately track and predict the movement of dynamic objects across frames. This temporal understanding allows us to more effectively filter out features associated with moving objects, resulting in a more stable and accurate SLAM process in dynamic environments.

3.4 Integration of ORB-SLAM2, Mask R-CNN, and SAM 2

Our proposed system integrates these three components to create a more robust SLAM solution for dynamic environments. This integration takes the form of three pre-processing steps that are enacted before frames are provided to the core ORB-SLAM 2 process. The pre-processing pipeline of frames is shown in a flowchart in Figure 4, with detailed descriptions below:

- **Mask Initialization:** When a frame is labeled as a keyframe by the ORB-SLAM 2 process, it is processed by a Mask R-CNN model, which generates masks for dynamic objects detected in the scene. A "dynamic object" is defined as any object labeled with a semantic class from a predefined set (e.g., person, truck, motorcycle) that is likely to

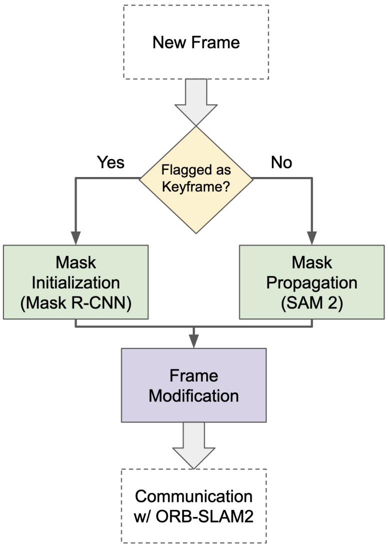

Figure 4: Pre-processing pipeline for dynamic object removal, performed before ORB-SLAM 2 uses the frame for localization.

move, with a confidence score above 0.9. The Mask R-CNN model used in this project was pretrained on the COCO (Common Objects in Context) dataset and is available through the Torchvision library. The dynamic object masks from the keyframe are saved for use in subsequent intermediary frames.

- **Mask Propagation:** When a frame is not labeled as a keyframe by the ORB-SLAM 2 process, it is processed by the SAM 2 model. SAM 2 enables the propagation of predefined masks across a sequence of frames by using the initial mask as a prompt and performing feature matching to track the masked object in subsequent frames. This allows non-rigid objects to be tracked as dynamic objects, with the mask adapting to changes in the object's shape. In our system, SAM 2 is initialized with a pair of frames—the local keyframe and the intermediary frame of interest. The keyframe masks are used as the initial prompt, and SAM 2's propagation feature is then used to predict the masks for the new frame. These predicted masks are then passed as the dynamic object masks for the frame.

- **Communication with ORB-SLAM:** Dynamic object masks are generated from each frame, either through initialization with Mask R-CNN or propagation with SAM 2. Dynamic objects in the scene are then marked black to remove their features. The processed frame is subsequently sent to the ORB-SLAM 2 pipeline. In our experiments, we used ROS 2 (Robot Operating System 2) for communication, with mask generation and ORB-SLAM running on separate nodes. ROS 2 was chosen due to the different programming languages used (SAM 2 is in Python, while ORB-SLAM 2 is implemented in C++), and because the project team was already proficient with ROS-based communication.

4 Experiment

4.1 Reference Videos

To evaluate our system in comparison with ORB-SLAM 2, we selected an RGB-D dataset from the Technical University of Munich (TUM) (Sturm et al., 2012), specifically designed for SLAM system evaluation. Although the dataset includes depth information for the relevant videos, we opted to use only the monocular videos due to memory constraints on the laptops used for testing. To effectively compare the performance of our system and ORB-SLAM 2, we focused on four video categories, with representative frames shown in Figure 5:

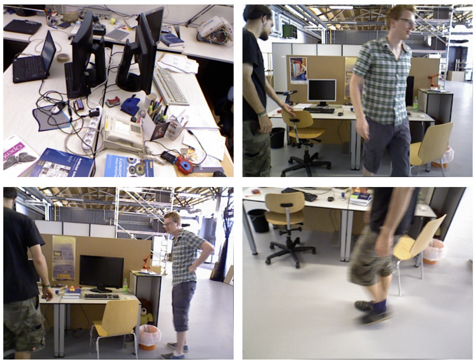

Figure 5: Representative frames from reference videos used. Video 1 (top left) is a static scene. Video 2 (top right) is a dynamic scene with a static camera. Video 3 (bottom left) is a dynamic scene with a translating camera. Video 4 (bottom right) is a dynamic scene with a rotating camera.

- **Reference Video 1: (freiburg 1 desk)** A static scene with camera movement, chosen to evaluate SLAM in a scenario where the baseline ORB-SLAM 2 is typically well-suited.

- **Reference Video 2: (freiburg 3 walking static)** A dynamic scene with a static camera and two people walking through the scene.

This video allows us to compare the performance of both SLAM systems in a scenario with limited views for localization.

- **Reference Video 3: (freiburg 3 walking xyz)** A dynamic scene with camera translation in all three axes (XYZ) and two people walking. The goal is to assess the robustness of both SLAM systems to camera translation.

- **Reference Video 4: (freiburg 3 walking rpy)** A dynamic scene with a rotating camera (roll, pitch, and yaw) and two people walking. This video helps evaluate the robustness of both SLAM systems to camera rotation.

4.2 Analysis

The analysis of the results presented in Table 1 demonstrates that our integrated approach of ORB-SLAM 2 with Mask R-CNN and SAM 2 offers significant improvements over the baseline ORB-SLAM 2 system, particularly in challenging dynamic environments.

For Reference Video 1, which represents a static scene, both systems performed comparably. Our integrated approach showed a slight increase in Mean RPE (410.1 vs 400.0) but with lower standard deviation (28.06 vs 31.02), indicating more consistent performance. Notably, our system generated more map points (4790 vs 3384), suggesting a more detailed reconstruction of the environment.

The most striking improvements are evident in Reference Videos 2 and 3, where the baseline ORB-SLAM 2 failed to produce any results (indicated by N/A). These scenarios involve dynamic objects and camera movement, which typically challenge traditional SLAM systems. Our integrated approach, however, successfully tracked the camera pose and generated map points in both cases, demonstrating its robustness in handling dynamic scenes.

For Reference Video 4, involving a rotating camera in a dynamic scene, both systems produced results. However, our approach generated significantly more map points (1952 vs 108), indicating a more comprehensive scene reconstruction. While the Mean RPE is slightly higher for our system (402.5 vs 367.9), the ability to maintain tracking and mapping in this challenging scenario is a notable achievement.

The consistent mean processing time of 0.21s for our system in dynamic scenes (Videos 2, 3, and 4) suggests that the additional computational load from Mask R-CNN and SAM 2 is manageable and does not significantly impact real-time performance.

In conclusion, our integrated approach demonstrates considerable improvements in handling dynamic scenes and maintaining tracking where the baseline system fails. This robustness in challenging environments is crucial for real-world SLAM applications, making our system a valuable advancement in the field.

5 Conclusion

Our research has demonstrated the effectiveness of integrating advanced computer vision techniques, specifically Mask R-CNN and SAM 2, with the ORB-SLAM2 framework to create a more robust SLAM system capable of handling dynamic environments. The experimental results show significant improvements in several key areas:

- **Resilience in Dynamic Scenes**: Our system successfully maintained tracking and mapping in scenarios where traditional ORB-SLAM2 failed, particularly in environments with moving objects and camera motion.

- **Improved Map Quality**: The integrated approach consistently generated more map points across various scenarios, indicating a more comprehensive and detailed reconstruction of the environment.

- **Consistent Performance**: While there was a slight increase in Mean Relative Pose Error (RPE) in some cases, our system showed more consistent performance with lower standard deviations in RPE.

- **Real-time Capability**: Despite the additional computational load of Mask R-CNN and SAM 2, our system maintained reasonable processing times, demonstrating its potential for real-time applications.

These improvements address a critical limitation of traditional SLAM systems, making our approach more suitable for real-world applications where dynamic objects are common. The ability to effectively handle moving objects opens new possibilities in various domains, including: Autonomous navigation in urban environments, Robotic assistance in crowded spaces, Augmented reality in non-static scenes, Intelligent surveillance systems and Interactive mapping for smart city applications.

ORB-SLAM 2					**ORB-SLAM 2 + Mask-R-CNN + SAM 2**			
Ref. Video	**Mean RPE**	**Stdev RPE**	**Mean Time**	**Map Pts.**	**Mean RPE**	**Stdev RPE**	**Mean Time**	**Map Pts.**
Video 1	400.0	31.02	0.04s	3384	410.1	28.06	0.15s	4790
Video 2	N/A	N/A	N/A	N/A	422.6	24.08	0.21s	283
Video 3	N/A	N/A	N/A	N/A	402.0	40.43	0.21s	777
Video 4	367.9	23.00	0.04s	108	402.5	53.92	0.21s	1952

Table 1: Results of Base ORB-SLAM 2 and ORB-SLAM 2 with Integrated Mask-R-CNN + SAM 2.

The proposed methodology showcases the potential of integrating deep learning techniques with traditional SLAM frameworks. By leveraging the object detection capabilities of Mask R-CNN and the temporal tracking abilities of SAM 2, we have demonstrated a novel approach to addressing the fundamental challenge of dynamic object handling in simultaneous localization and mapping.

6 Future Work

Future research directions include:

- Develop more efficient integration techniques to reduce computational overhead
- Extend system evaluation across diverse and more complex environments
- Investigate semantic mapping capabilities
- Explore multi-sensor fusion approaches
- Implement adaptive dynamic object handling mechanisms

These focused research paths aim to transform our current approach into a more generalized and robust solution for dynamic scene understanding in robotics and computer vision applications.

References

Berta Bescos, José M Fácil, Javier Civera, and José Neira. 2018. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083.

Zheng Fang, Shibo Zhang, Yi Wang, and Zhongxiang Wang. 2019. Deep learning for visual slam in dynamic environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8290–8297.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969.

Raúl Mur-Artal and Juan D Tardós. 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. In *IEEE Transactions on Robotics*, volume 33, pages 1255–1262.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. 2012. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*.

Yuxin Wang, Zeyu Sun, Chi-Zhi Xu, and Yue Liu. 2022. Robust dynamic slam with semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1000–1009.

Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. 2018. Ds-slam: A semantic visual slam towards dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174.

Tao Yuan, Huimin Hu, Jiawei Fu, and Liang Guo. 2022. Segmap: 3d segment mapping using data-driven descriptors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3846–3855.

A Appendix

Reference Video 3 demo of both of our approach and base ORB-SLAM can be accessed at the following link: https://drive.google.com/drive/folders/14WdSg8yw4wgoomxi0eUZVjofjPttnKxF. (The base ORB-SLAM video is in real-time, while our approach is sped up to 6x speed)

Our GitHub code bases can be found here:
https://github.com/kn1451j/DynORBBase
https://github.com/kn1451j/DynORB