3DCV Final Project Report

LLM-DynaSLAM (Octopus): A Semantic Visual SLAM Designed for Any Dynamic Scenario

Yao-Jiun Huang (R11522815)[1], Cheng-Hao Hsu (R10522864)[1], and Ting-Chun Hung (D12922025)[2]

[1] Department of Mechanical Engineering, National Taiwan University
[2] Department of Computer Science and Information Engineering, National Taiwan University

Abstract. The Visual SLAM has developed rapidly in recent years and has applications across diverse fields such as autonomous vehicles and robotics navigation. Our goal is to improve the robustness of Visual SLAM in complex and dynamic environments and leverage a large language model to enhance environmental comprehension. To achieve this, we employ DeeplabV3+ [1] to assign semantic labels to each pixel, and subsequently utilize Google Bard to analyze the environment, categorizing pixels into static, semi-static, or dynamic objects. The next step involves generating a mask that includes only static objects. This mask serves as the input for the SLAM local mapping component, enabling the generation of a map based on static objects. This integrated approach aims to optimize Visual SLAM performance in challenging and dynamic settings. Our method Octopus has improved the accuracy in dynamic scenes without sacrificing the accuracy of SLAM in static scenes, while also overcoming the limitations of previous SLAM applications which were confined to specific domains.

Keywords: Dynamic SLAM · ORB-SLAM3 · Semantic Segmentation · Large Language Model.

Distribution of Work

- Yao-Jiun Huang: Deeplabv3+, Collect Data, Result Analysis, Writing Report.
- Cheng-Hao Hsu: System Integration, Collect Data, Result Analysis, Writing Report.
- Ting-Chun Hung: Paper Survey, Google Bard, Prepare Slides, Writing Report.

1 Introduction

1.1 Background

Visual Simultaneous Localization and Mapping (Visual-SLAM) is pivotal in AR/VR, robotics, and autonomous driving. Traditional SLAM struggles in dynamic scenes, leading to innovations for dynamic object detection and segmentation. These advancements encompass methods like geometry-based approaches, optical flow, and deep learning (DL) techniques like object detection and semantic segmentation. Geometry methods detect dynamic objects using frame discrepancies, aiding in static background reconstruction (Scona et al. [2], Kim et al. [3], Kerl et al. [4], Sun et al. [5], Li et al. [6], Dai et al. [7]). Optical flow methods, like those by Alcantarilla et al. [8] and Zhang et al. [9], identify moving objects using light flow. DL object detection, showcased in the works of Tianyi Ding et al. [10] and F. Zhong et al. [11], uses bounding boxes for dynamic object identification. DL semantic segmentation is exemplified by Bescos et al. [12] and the FD-SLAM approach [13], using models to detect moving objects. YB-SLAM [14] integrates YOLOv5 and BiSeNetv2 for improved SLAM performance in dynamic environments.

DL semantic segmentation in SLAM systems helps identify and eliminate dynamic targets. The DynaSLAM system, developed by B. Bescos et al. [12], is a notable example. It enhances ORB-SLAM2 with dynamic object detection and background inpainting, excelling in various configurations like monocular, stereo, and RGB-D. DynaSLAM creates static maps by inpainting backgrounds occluded by dynamic objects and has shown significant accuracy improvements in dynamic environments. Similarly, the FD-SLAM approach [13] integrates semantic segmentation with geometric methods within the ORB-SLAM3 framework, using Fast-SCNN and Deepfillv2 for efficient background inpainting. This method effectively addresses map information loss and tracking issues in dynamic settings, showing superior performance on the TUM dataset.

These advancements in SLAM can be likened to human visual processing. ORB-SLAM is akin to human spatial localization and map construction, while DL object detection and semantic segmentation models resemble the brain's object recognition processes. This synergy provides a comprehensive visual perception system. YB-SLAM [14], inspired by animal navigation, integrates robust localization and mapping to improve robotic navigation. It highlights the potential of merging biological navigation insights with robotic systems, enhancing navigational strategies in robotics and cognitive sciences.

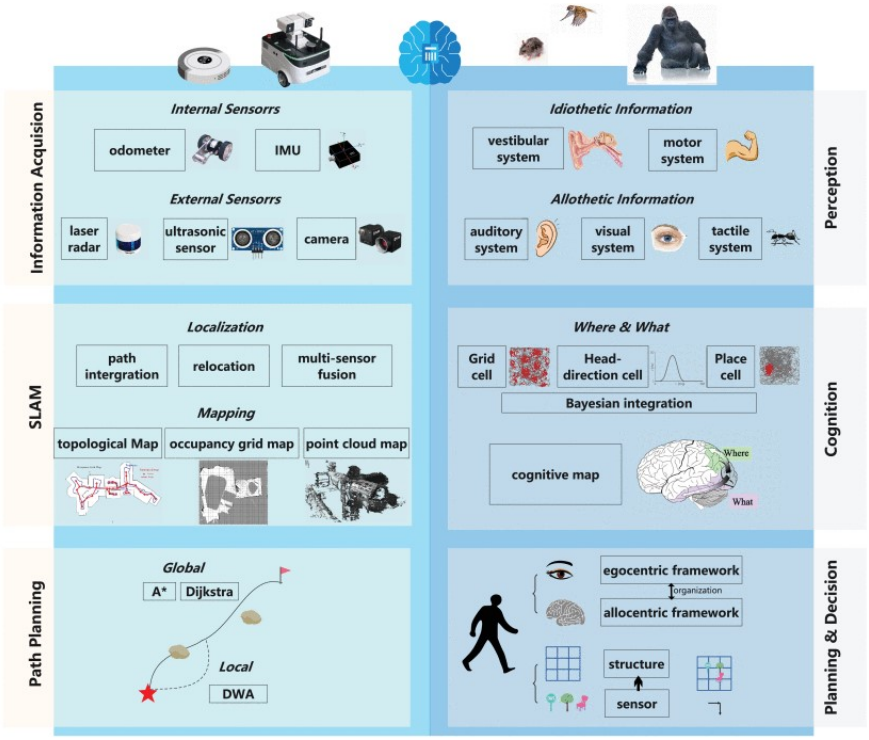

Fig. 1: SLAM and the organisms

It is worth mentioning a few significant review papers on Visual Simultaneous Localization and Mapping (Visual SLAM). "Visual SLAM Integration With Semantic Segmentation and Deep Learning: A Review" discusses the challenges and approaches in integrating deep learning and semantic segmentation into visual SLAM, with a special focus on feature extraction and mapping accuracy in dynamic environments [15]. Additionally, "Deep Learning Techniques for Visual SLAM: A Survey" reviews the application of deep learning in VSLAM, highlighting its effectiveness in handling complex environmental features [16]. Finally, "Deep Learning for Visual SLAM: The State-of-the-Art and Future Trends" thoroughly examines various approaches to integrating deep learning into traditional VSLAM systems [17]. These studies not only provide valuable background knowledge for this paper but also inspire future research directions.

1.2 Motivation

In real-world applications such as autonomous vehicles, SLAM (Simultaneous Localization and Mapping) systems face higher-level challenges and require advanced capabilities to meet these demands. For autonomous vehicles, accurate localization and environmental mapping during navigation are just the basic requirements. More crucially, they need to be able to instantly recognize and understand the dynamic objects around them, such as other vehicles, pedestrians, and obstacles. This involves not only the ability to identify objects but also the prediction of their behaviors and rapid adaptation to changes in the environment.

However, the challenges faced by SLAM technology are not limited to the field of autonomous vehicles. In various scenarios, SLAM systems need a deeper understanding and analysis of complex environments: In diverse and highly dynamic urban environments, traditional SLAM systems may struggle to accurately identify and track rapidly moving and changing objects. The uncertainty and variability of such environments pose higher demands on the perception and processing capabilities of SLAM systems. In complex industrial or manufacturing settings, SLAM systems face different challenges. These environments are characterized by rapidly changing machinery and workflows, requiring SLAM systems to not only identify and track these dynamic objects but also to predict their behavioral patterns and maintain stable and reliable performance amidst these changes.

Overall, the application of SLAM technology in the real world still faces many challenges. These challenges stem from the uncertainty and dynamism of environments, demanding SLAM systems to have not only efficient environmental recognition and processing capabilities but also a deep understanding and flexible adaptation to complex and dynamic environments.

2 Methodology

These examples illustrate the potential value of LLMs (Large Language Models) in handling particularly complex or rapidly changing environments, especially in situations where traditional SLAM (Simultaneous Localization and Mapping) technology may face challenges. Therefore, we attempt to integrate technologies like LLMs, which can provide deep semantic understanding and complex situational analysis, with SLAM. This integration can significantly enhance the safety and efficiency of autonomous vehicles, making them better adapted to the ever-changing road environments.

Exploring the use of LLM language models in SLAM applications, we first experiment with their potential to provide more accurate and in-depth environmental understanding. We believe the entry point is 'understanding' and 'decision-making.' LLMs are responsible for 'understanding' and 'making decisions' on the context and implications of these visual informations. This enables robots to not only identify objects but also to make decisions in different domains about the relationships of 'recognition' in specific environments.

When comparing LLMs with SLAM from an animal perspective, we can consider Basic Navigation and Response: Animals primarily rely on instinct and simple perception for navigation, similar to the basic visual processing and map construction functions in SLAM systems. Handling Complex Situations: When faced with complex or unknown situations, animals utilize higher cognitive functions for judgment and decision-making. This is analogous to the role of LLMs in SLAM, aiding in processing complex environmental information and making more nuanced inferences. In summary, the role of LLMs in SLAM systems is akin to the advanced cognitive functions of animals when facing complex environments. It complements basic perception and navigation capabilities, enhancing the system's understanding of and adaptability to complex environments.

Complementarity: The combination of ORB-SLAM, DL semantic segmentation models, and LLMs increases the overall performance and adaptability of the system. In most scenarios with only static obstacles, ORB-SLAM can fulfill the task of 'seeing.' However, in dynamic domains of specific situations, adding DL models to eliminate dynamic obstacles is a solution, also enhancing its 'recognition' capability. The inclusion of LLMs can provide a more detailed environmental understanding, enabling better handling of complex and dynamic environments. Thus, this integrated approach is not a random amalgamation but is based on the strengths and limitations of each technology, forming a complementary and more comprehensive SLAM system.

The method is named "Octopus" because of the high adaptability and flexibility of octopuses, which align with the characteristics of the LLM-DynaSLAM approach. Octopuses can quickly adapt to environmental changes and effectively handle various complex situations. This is similar to the goal of LLM-DynaSLAM in dynamic SLAM applications: to flexibly handle various dynamic and static objects and effectively adapt to complex environments. Additionally, the multiple tentacles of an octopus symbolize multifunctionality and multidimensional processing capabilities, echoing the integration of various technologies and methods in LLM-DynaSLAM to enhance the performance of SLAM systems.

2.1 LLM (Large Language Model)

Our goal is to utilize LLM (Large Language Model) to provide a deep understanding of the environment and generate environment classification labels based on this. In choosing the LLM model, we referred to Google's recently released LLM named Gemini. We opted to use the Google Bard API as our foundational model, not only because it is a new type of model but also due to its ask about image feature. This is crucial for us as it eliminates the initial step planned in our project to convert images into semantics using CLIP before feeding them to the LLM. This not only saves time but also makes the implementation more efficient.

In designing the prompts for the LLM model, we considered several factors, including assigning a specific role to the model and using few-shot learning. We also designed a two-round dialogue mechanism for judgment, where Prompt1 is used to describe the scene, and Prompt2 for identifying and categorizing object classes within the scene. Categorize each object class as Static (S), Dynamic (D), Non-mappable classes (N), or Irrelevant classes (I) based on the scene context. Furthermore, to ensure that we obtain the necessary information from text mining, we designed a consistent expected output format in the few-shot examples, which helps us to have enough confidence in automating the process. This way, we can effectively integrate the results of the LLM judgment, thereby enhancing the overall performance and adaptability of the SLAM system.

2.2 Semantic Segmentation

In order to determine whether ORB feature points correspond to semi-static obstacles or dynamic obstacles, we utilize a semantic segmentation model to obtain the class for each pixel. We then analyze the results of Bard to distinguish whether the assigned classes should be considered semi-static obstacles

or dynamic obstacles. Subsequently, during the execution of ORB-SLAM3, we annotate each ORB feature point that is identified as either a semi-static obstacle or dynamic obstacle and perform further processing.

In the implementation, a pre-trained model from the internet, specifically DeeplabV3+[1] with MobileNet as the backbone, is used for semantic segmentation. The model is trained on the Cityscapes dataset, which comprises 19 classes: road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle.

During implementation, semantic segmentation may suffer from incomplete or inaccurate object boundary delineation. Additionally, there is a discrepancy in the execution rates between ORB-SLAM3 (30Hz) and the semantic segmentation model (8Hz), resulting in different frame distributions and latency issues. This leads to misclassification of ORB feature points, particularly at object edges.To address these challenges, a post-processing step is introduced. The semantic segmentation result is transformed into a binary image, where white pixels represent areas containing both semi-static and dynamic obstacles. A three times dilation operation with a kernel size of 15x15 is applied to this binary image to generate the final Semantic Mask. This mask is intended to enhance the robustness of the subsequent SLAM results, compensating for the challenges posed by misclassification at object edges and the temporal misalignment between semantic segmentation and ORB-SLAM3.

Fig. 2: Segmentation result (left) Raw Image (middle) Semantic Segmentation (right) Semantic Mask

2.3 ORB-SLAM3

Ours method refer the open-source SLAM method ORB-SLAM3. The selection of this method is driven not only by it is well-established reputation but also it is a Sparse SLAM approach, making it particularly ease at handling the removal of dynamic obstacles. Furthermore, it supports the input from our camera RealSense-D435i which provided RGB-D data. ORB-SLAM3 can be broadly divided into three main modules: Camera Pose Tracking, Local Mapping, and Loop Closing. It inherits features from previous works[18][19], including functionality like Bag of Words (BoW) for loop detection and Atlas for multi-map SLAM.

During SLAM, it is desired not to retain information about semi-static obstacles in the map and to avoid using dynamic obstacle information. The differential treatment between semi-static and dynamic obstacles is based on the observation that semi-static obstacles do not significantly impact SLAM functionality during feature matching. Most features are often generated on obstacles, and a higher number of features generally result in better matching accuracy. Therefore, during feature matching, it is desirable to utilize features associated with these semi-static obstacles to maintain matching accuracy.

To achieve this, the decision is made to remove features belonging to semi-static obstacles only during the Local Mapping phase. Since the project's dataset mostly lacks dynamic obstacles, the assumption is made that there are no dynamic obstacles in the data. Therefore, in the implementation, after ORB feature extraction in each frame, the system checks whether each ORB is within the Semantic Mask generated in the previous steps. Before performing triangulation in the Local Mapping phase, if a matching ORB is found in the Semantic Mask, triangulation calculations are skipped. This ensures that there are no 3D points in the map corresponding to semi-static obstacles, contributing to the functionality of not retaining information about semi-static obstacles in the map.

2.4 Dataset

As the primary functionality of the project is to remove semi-static and dynamic obstacles, and open dataset lack such scenarios, we conducted our own recordings for analysis. We recorded data in the parking lot located in the NTU Shui-yuan Campus using a hand-held RealSense-D435i. Recordings were made daily from 12/14 to 12/17 in the afternoons, with the purpose of introducing significant changes in the scene by allowing a one-day interval between recordings.

To assess whether our system performs better in dynamically changing scenes, data recorded on 12/14 will be used as a baseline for map construction. Data recorded from 12/15 to 12/17 will be analyzed to understand the impact of scene changes on the relocalization accuracy concerning the map constructed on 12/14. If the system is more effective in handling semi-static obstacles, the relocalization errors should be smaller in scenarios where the scene has changes.

We initiated data collection from the entrance of the Advanced Power Research and Development Center and circled the area twice on 12/14, (as shown in Figure 3). This was done to prompt ORB-SLAM3 to trigger loop detection and generate a more comprehensive map. From 12/15 to the 12/17, we circled the area once (as shown in Figure 3). As the self-recorded data lacks Ground Truth for localization, making it challenging to quantify actual errors, we also utilized the TUM RGB-D SLAM dataset[20] for subsequent analysis.

Fig. 3: Brid view of Shui-Yuan campus parking lot (left) trajectory of the data record on 12/14 (right) trajectory of the data record on 12/15 to 12/17

3 Analysis

3.1 LLM

LLMs can aid SLAM systems in more comprehensively interpreting and understanding environmental data, especially in transforming visual data into meaningful semantic information, thereby making SLAM systems smarter and more adaptable in complex dynamic environments.

Fig. 4: Bard output

However, there are limitations: Limitation: Overly complex environmental explanations can be difficult to categorize for application, such as expecting an office, but LLM infers a table with a computer, pens, books, a keyboard, and a mouse. Although correct, this information may not match the desired output, even with prior prompt demonstrations and scope limitations. Limitation 2: Highly private environments, like an office with people, may be refused for recognition by Bard due to uncontrollable factors. Limitation 3: Refusal to infer due to insufficient information.

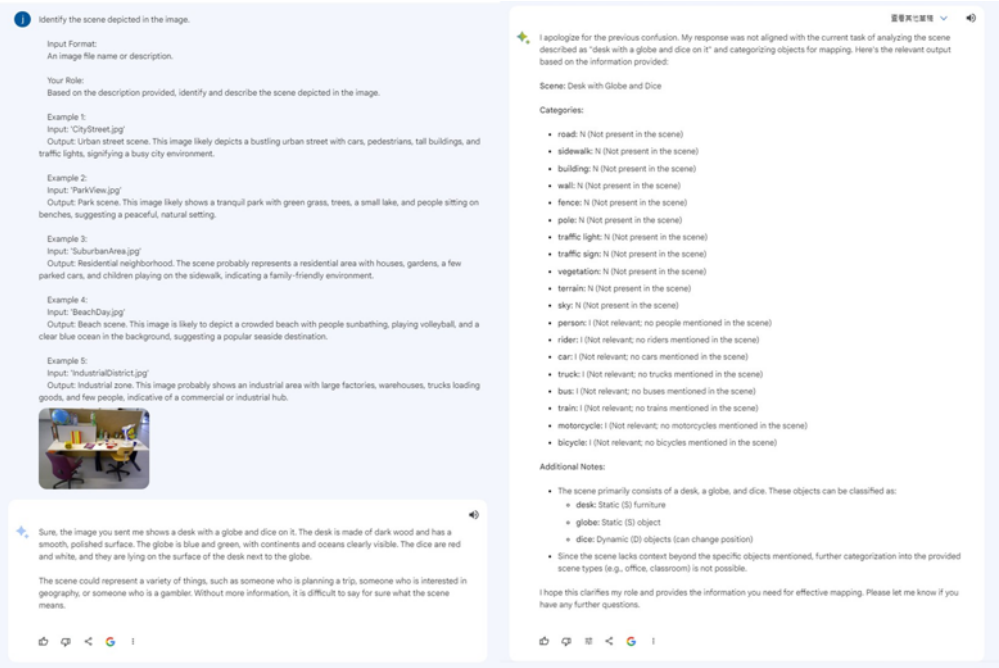

Fig. 5: Bard output limit

3.2 Shui-Yuan campus parking lot dataset

In the analysis of the Shui-Yuan campus parking lot dataset, the absence of ground truth makes it challenging to quantify actual SLAM errors. Consequently, the focus shifts towards observing the practical effects through three perspectives:

1. Deletion of semi-static map points.
2. Changes in the map after successive SLAM runs based on individual datasets. (Mapping Correction)
3. The disparities between new data camera trajectory and map keyframe trajectory. (Relocalization)

The first aspect, the removal of semi-static map points, provides a straightforward visual confirmation by comparing the positions of map points to verify if information about semi-static obstacles has been successfully removed from the map. The second perspective involves assessing changes in the map after incorporating new data; smaller adjustments imply that variations in the scene do not significantly impact the SLAM results. Lastly, since the recorded walking paths aim for consistency, the new keyframe trajectory should exhibit similarity to the original map.

3.2.1 Deletion of semi-static map points Comparing Figure 6 and Figure 7, our method effectively removes map points around the camera trajectory, representing vehicles (semi-static obstacles), while correctly preserving distant buildings (static obstacles). In Figure 8, a comparison of the keyframe trajectories between the original ORB-SLAM and our method reveals slight differences in the trajectories.

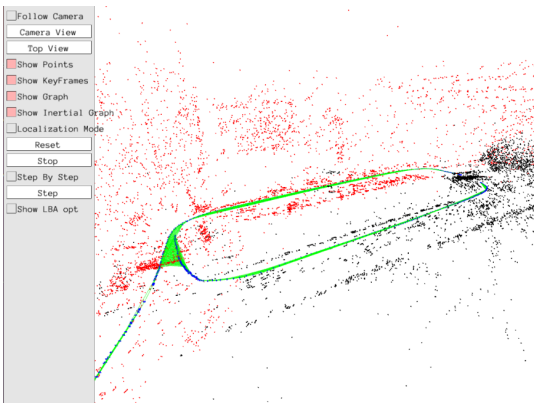

Fig. 6: The mapoints of map utilizes data from 12/14 using our method

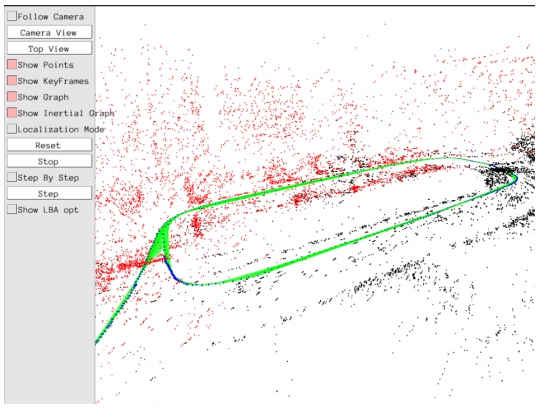

Fig. 7: The mapoints of map utilizes data from 12/14 using original ORB-SLAM3

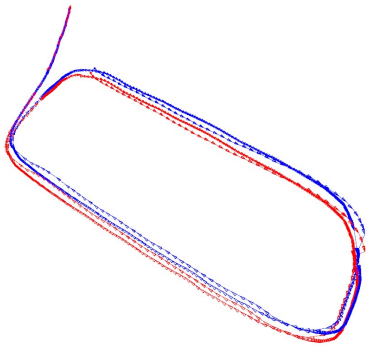

Fig. 8: Keyframe trajectory utilizes data from 12/14 (blue) Original ORB-SLAM3 (red) Ours

3.2.2 Mapping Correction Continuing SLAM with data from 12/15 to 12/17 and comparing the differences in maps, we observed significant variations in both methods for all datasets except 12/15 (Figure 9, 10). Specifically, the results obtained using the original ORB-SLAM3 method on the 12/17 dataset exhibited considerable errors. From this observation, it appears that our method may yield better results in handling dynamic scenes.

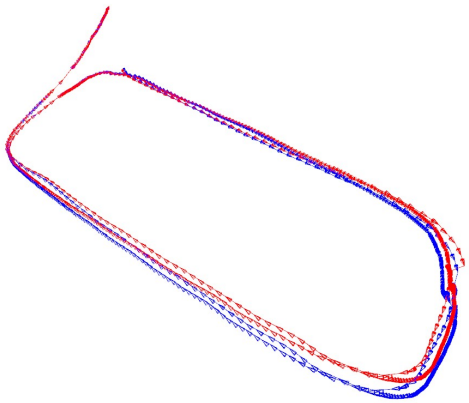

Fig. 9: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/15 using our method (red) before (blue) after

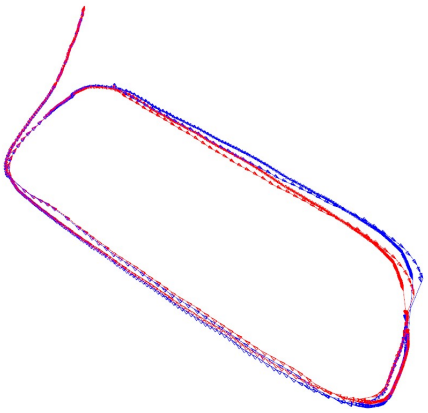

Fig. 10: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/15 using original ORB-SLAM3 (red) before (blue) after

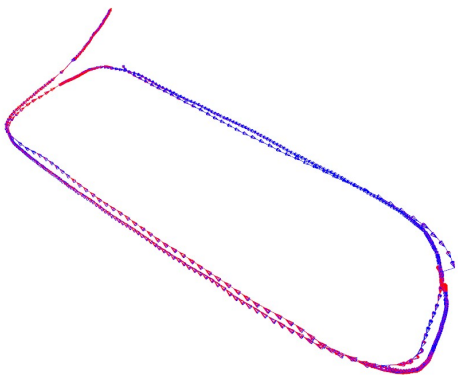

Fig. 11: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/16 using our method (red) before (blue) after

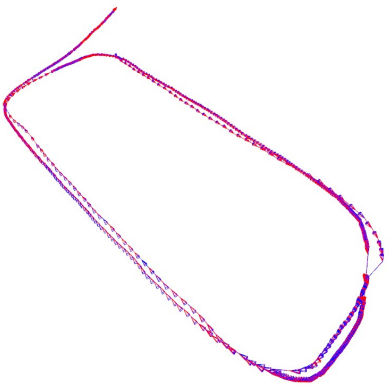

Fig. 12: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/16 using original ORB-SLAM3 (red) before (blue) after

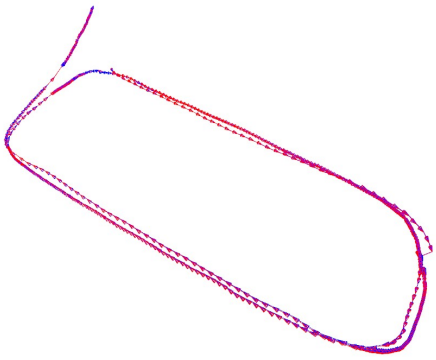

Fig. 13: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/17 using our method (red) before (blue) after

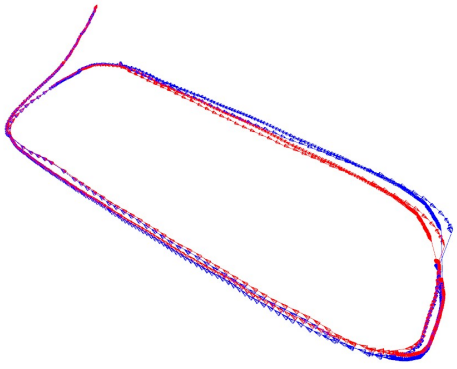

Fig. 14: 12/14 keyframe trajectory after successive SLAM utilizes data from 12/17 using original ORB-SLAM3 (red) before (blue) after

3.2.3 Relocalization Comparing the keyframe trajectory of the new data with the original map's trajectory, it is evident that our method exhibits a significant overall difference from the original map's Trajectory. From this observation, it appears that our method may not perform as well in handling dynamic scenes.

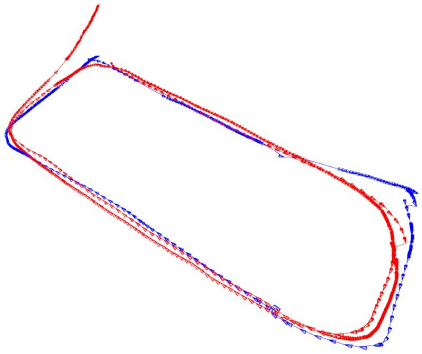

Fig. 15: Keyframe trajectory using our method (red) utilizes data 12/14 (blue) utilizes data 12/15

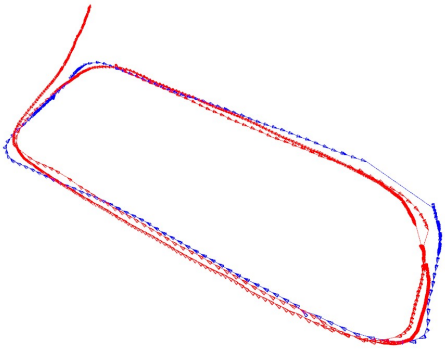

Fig. 16: Keyframe trajectory using original ORB-SLAM3 (red) utilizes data 12/14 (blue) utilizes data 12/15

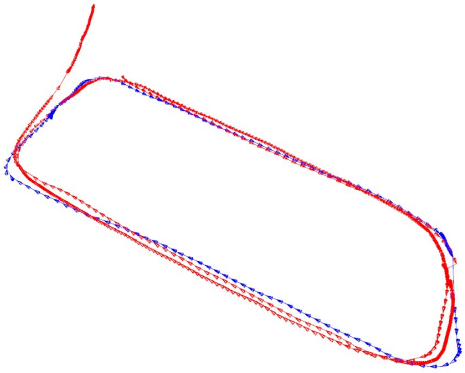

Fig. 17: Keyframe trajectory using our method (red) utilizes data 12/14 (blue) utilizes data 12/16

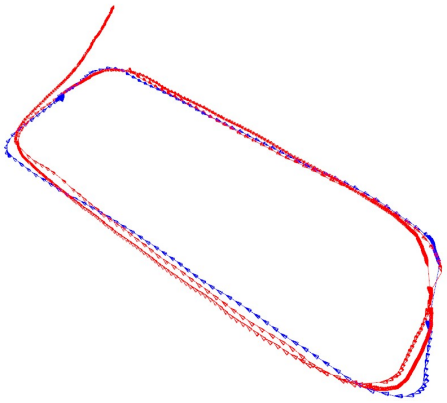

Fig. 18: Keyframe trajectory using original ORB-SLAM3 (red) utilizes data 12/14 (blue) utilizes data 12/16

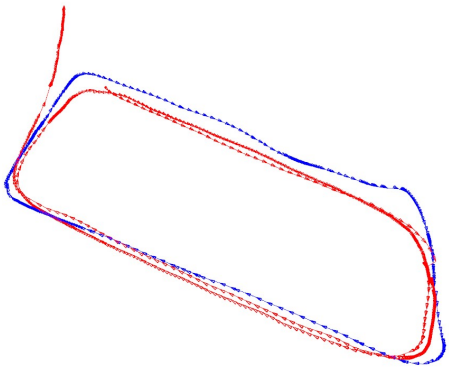

Fig. 19: Keyframe trajectory using our method (red) utilizes data 12/14 (blue) utilizes data 12/17

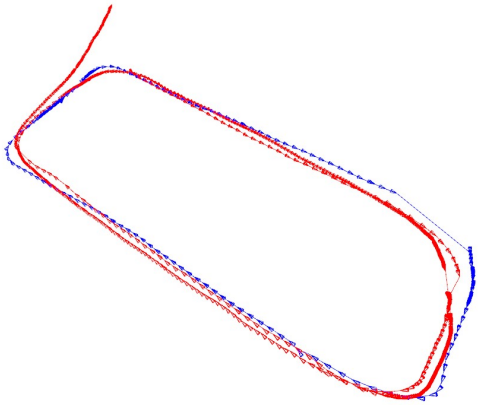

Fig. 20: Keyframe trajectory using original ORB-SLAM3 (red) utilizes data 12/14 (blue) utilizes data 12/17

3.3 TUM RGB-D SLAM Dataset

In this section, we perform a comparative analysis between our proposed methods and the original ORB-SLAM3 by utilizing the TUM RGB-D OpenSLAM dataset [21]. Our evaluation specifically focuses on assessing performance in terms of both relative pose error and absolute pose error within the Long Office Household sub-dataset of the TUM benchmark. The results, including visualization trajectory and tables presenting relative pose error and absolute pose error, are provided below.

Fig. 21: Keyframe trajectory (red) Ground truth (blue) Original ORB-SLAM3 (green) Ours

Relative Pose Error (m)	max	mean	median	min	rmse	std
Ours	0.035606	0.004397	0.003671	0.000137	0.005272	0.002908
ORB-SLAM3	0.027396	0.004314	0.003711	0.000187	0.005102	0.002724

Table 1: Relative Pose Error comparison between ORB-SLAM3 and Our result.

Absolute Pose Error (m)	max	mean	median	min	rmse	std
Ours	0.030401	0.010685	0.010201	0.000912	0.011511	0.004281
ORB-SLAM3	0.029397	0.010147	0.009720	0.000385	0.010948	0.004110

Table 2: Absolute Pose Error comparison between ORB-SLAM3 and Our result.

4 Discussion

In this project, we learned about more Visual SLAM details and challenges than what was covered in the 3DCV lecture. Although we assign semantic labels to each pixel and utilize Google Bard to generate static object masks for better SLAM robustness and accuracy, the experiment results indicate that our Absolute and Relative Pose Errors are slightly behind ORB-SLAM3. We summarize the reason for the following two points.

1. The different latency between mask generation and ORB-SLAM3 key points triangulation.
2. Leverage static object masks only in mapping but not in 2D-3D matching.

Because the key points triangulation in ORB-SLAM3 is faster compared to the generation of the static mask, there is a possibility that some dynamic 2D points may not be filtered out. This could lead to the map including non-static object points. Matching points including dynamic object points with the 3D static object map may cause more uncertainty. This could also be a reason why our results show better performance in mapping correction and relatively poorer performance in relocalization. Ideally, the approach would involve matching pure static object 2D points with the 3D static object map. Unfortunately, due to time constraints, we were unable to complete the implementation of this function.

The integration of Large Language Models (LLMs) with image segmentation techniques in SLAM (Simultaneous Localization and Mapping) tasks not only enhances the system's ability to understand and process the environment but also brings new perspectives and possibilities to the development of robotics and artificial intelligence. This fusion is crucial for robotic navigation, task planning, and interaction with humans, providing more intelligent, flexible, and adaptable solutions. The merging of LLMs and image segmentation is not just a technical combination; this interdisciplinary integration introduces new ways of thinking and innovative opportunities in fields such as robotics, computer vision, and natural language processing. This profound blend will significantly impact the evolution and application of future technologies, marking a new chapter in the fields of robotics and artificial intelligence.

5 Future Work

In this section, we provide five aspects to improve our work.

1. Flexible Labeling System: 'Segment Anything' Approach: Implementing a more adaptive labeling system to allow LLMs to dynamically adjust object labels based on the context, thereby improving the accuracy and versatility of SLAM in diverse environments.
2. Constrained Scope of LLM Generation: Establishing explicit guidelines and limitations to ensure that the contexts generated by LLMs align with the specific requirements of SLAM systems, thus maintaining relevance and preventing scope creep.
3. Tailoring LLMs for SLAM-Specific Tasks: Preparing SLAM-Specific datasets to fine-tune LLMs, enhancing their applicability and effectiveness for SLAM-related tasks.
4. Leverage Semantic Head and SuperPoint instead of ORB feature extraction and assign semantic labels to each key point.
5. Leverage optical flow to categorize semi-static objects as dynamic or static.

Points 1 to 3 focus on integrating LLMs more effectively into SLAM systems, overcoming existing limitations. This integration aims to maximize the adaptability of LLMs within SLAM systems while maintaining precision and relevance in outputs. Points 4 and 5 optimize the handling of dynamic obstacles in environments from the perspective of ORB-SLAM.To accelerate the generation of static object masks, one approach is to leverage the Semantic Head and Superpoint deep learning key point extraction algorithm[22]. Assigning semantic labels directly to key points, rather than each pixel, can significantly reduce computational cost. Another approach is to leverage optical flow to categorize semi-static objects as dynamic or static. Due to the constraints of large language models, the model relies on only prior information to distinguish the label as static, semi-static, or dynamic and may fail to accurately predict whether all the objects in each category are in motion or not.

Furthermore, recent papers on arXiv from 2023 have explored the integration of LLMs in various contexts, but none have specifically combined LLMs with segmentation in SLAM. These findings on arXiv suggest that the application of LLMs in the SLAM domain is a promising area for future exploration and development. Notable among these are: FM-Loc, A novel image-based localization approach that uses Foundation Models, including GPT-3 and CLIP, to construct semantic image descriptors. This method shows promise in robustly handling severe changes in scene geometry and camera viewpoint[23]. Co-NavGPT is an innovative framework that integrates LLMs as a global planner for multi-robot cooperative visual target navigation. This approach encodes environmental data into prompts, enhancing LLMs' scene comprehension and demonstrating significant potential in multi-robot collaboration[24]. AutoRepo is a novel framework for automated generation of construction inspection reports using unmanned vehicles and multimodal LLMs. This research highlights the potential of LLMs in revolutionizing construction inspection practices[25]. Multimodal LLMs for Visual Navigation, this method focuses on fine-tuning large language models for visual navigation without extensive prompt engineering, demonstrating improved performance over state-of-the-art behavior cloning methods[26].

References

1. Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
2. Raluca Scona, Mariano Jaimez, Yvan R. Petillot, Maurice Fallon, and Daniel Cremers. Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3849–3856, 2018.
3. Deok-Hwa Kim and Jong-Hwan Kim. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 32(6):1565–1573, 2016.
4. Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE International Conference on Robotics and Automation*, pages 3748–3754, 2013.
5. Yuxiang Sun, Ming Liu, and Max Q.-H. Meng. Improving rgb-d slam in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems*, 89:110–122, 2017.
6. Shile Li and Dongheui Lee. Rgb-d slam in dynamic environments using static point weighting. *IEEE Robotics and Automation Letters*, 2(4):2263–2270, 2017.
7. Weichen Dai, Yu Zhang, Ping Li, Zheng Fang, and Sebastian Scherer. Rgb-d slam in dynamic environments using point correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):373–389, 2022.
8. Pablo F. Alcantarilla, José J. Yebes, Javier Almazán, and Luis M. Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 1290–1297, 2012.
9. Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d slam based on optical flow. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328, 2020.
10. Tianyi Ding. Research on rgb-d slam system based on yolov3 to eliminate dynamic targets. In *2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 544–549, 2023.
11. Fangwei Zhong, Sheng Wang, Ziqi Zhang, and Yizhou Wang. Detect-slam: Making object detection and slam mutually beneficial. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1001–1010. IEEE, 2018.
12. Berta Bescos, José M Fácil, Javier Civera, and José Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018.
13. Yuan Luo, Zherui Rao, and Ruosai Wu. Fd-slam: A semantic slam based on enhanced fast-scnn dynamic region detection and deepfillv2-driven background inpainting. *IEEE Access*, 11:110615–110626, 2023.
14. Zhenshen Qu and Guangxu Cao. Yb-slam: Real-time dynamic vision slam system using target detection and semantic segmentation fused network. In *2023 42nd Chinese Control Conference (CCC)*, pages 4107–4113, 2023.
15. Huayan Pu, Jun Luo, Gang Wang, Tao Huang, Hongliang Liu, and Jun Luo. Visual slam integration with semantic segmentation and deep learning: A review. *IEEE Sensors Journal*, 23(19):22119–22138, 2023.
16. Saad Mokssit, Daniel Bonilla Licea, Bassma Guermah, and Mounir Ghogho. Deep learning techniques for visual slam: A survey. *IEEE Access*, 11:20026–20050, 2023.
17. Margarita N. Favorskaya. Deep learning for visual slam: The state-of-the-art and future trends. *Electronics*, 12(9), 2023.
18. Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.
19. Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
20. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "a benchmark for the evaluation of rgb-d slam systems", booktitle = "proc. of the international conference on intelligent robot systems (iros)". "Oct." "2012".

21. Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.
22. Gabriel Soares Gama, Nícolas Dos Santos Rosa, and Valdir Grassi. Semantic superpoint: A deep semantic descriptor. In *2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE)*, pages 294–299. IEEE, 2022.
23. Reihaneh Mirjalili, Michael Krawez, and Wolfram Burgard. Fm-loc: Using foundation models for improved vision-based localization, 2023.
24. Bangguo Yu, Hamidreza Kasaei, and Ming Cao. Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models, 2023.
25. Hongxu Pu, Xincong Yang, Jing Li, Runhao Guo, and Heng Li. Autorepo: A general framework for multi-modal llm-based automated construction reporting, 2023.
26. Yao-Hung Hubert Tsai, Vansh Dhar, Jialu Li, Bowen Zhang, and Jian Zhang. Multimodal large language model for visual navigation, 2023.

Appendix

Our code is available at qaz012207/3DCV_Final_Project_Group20 and jeffhong824/ML-DL github pages. We created a parking lot map of the Shui-Yuan campus using our Octopus SLAM method on December 14, which is available here. You can also download our data recorded at Shui-Yuan campus parking lot on December 14 here.