Applications and Experiments on Query-based Object-aware Visual Language Mapping using Frame Division Method in Microcomputers

Jun Young Yun
Department of Artificial Intelligence
Korea Aerospace University
Goyang, Republic of Korea
a_o@kau.kr

Daeyoel Kang
Department of Artificial Intelligence
Korea Aerospace University
Goyang, Republic of Korea
kangdy1997@kau.kr

Noheun Myeong
Department of Artificial Intelligence
Korea Aerospace University
Goyang, Republic of Korea
2024320011@kau.kr

Jongyoon Park
Department of Artificial Intelligence
Korea Aerospace University
Goyang, Republic of Korea
jongyoon0621@kau.kr

Pileun Kim
Department of Autonomous Vehicle Engineering
Korea Aerospace University
Goyang, Republic of Korea
pkim@kau.ac.kr

***Abstract*—This paper demonstrates an approach to integrate object-aware map building technique which employs visual language models in microcomputers. This paper addresses the computational challenges of deploying visual language models in resource-constrained environments, such as mobile robots with microcomputers. It is achieved by separating object-aware map building process. The proposed mapping process is divided into 3 stages, data acquisition stage, object-aware map building stage, and inference stage. Experiments are conducted with Turtlebot4 mobile robot with Raspberry Pi microcomputers, to validate its performance in low-powered devices. The result showed 66% of success rate in overall text list, including 60% success rate with input texts with description. This study contributes to mobile robotics by showing that even microcomputers with limited processing power can support object-aware navigation tasks through optimized mapping method.**

Keywords—Object-aware navigation, Low-power devices, Visual language models, Real-time navigation, Cognitive robotics

I. INTRODUCTION

The emergence of Visual Language Models (VLMs) and Large Language Models (LLMs) has marked a significant advancement in the fields of visual recognition and language understanding, particularly in zero-shot inference. VLMs have showed remarkable performance in zero-shot visual labeling and general knowledge-based understanding without additional supervised training, which was traditionally necessary for conventional object detection or semantic segmentation models. Moreover, LLMs, trained with vast amount of internet-scale data, have achieved near-human performance in tasks related to textual question-answering and reasoning, also, capable of reasoning processes with human-like chain-of-thoughts.

These advancements suggest a future where robots can recognize and interpret the world with minimal prior training and perform reasoning like human cognition. However, deploying such heavy-weighted models on mobile robots involves significant challenges. Particularly when robots operate in resource-constrained environments without continuous internet connection. The computational demands of heavy-weighted VLMs or LLMs in real-time on mobile devices, such as microcomputers, are prohibitive, limiting their applicability in practical operations.

Our research aims to address these challenges by optimizing the use of VLMs within robotic mapping systems. We aim to make them feasible for real-time operations on low-powered microcomputers. In previous work, we developed a system that utilizes VLMs for object recognition during map-building processes. This system employed a division method to reduce complexity of the model. With experiment, the construction of object-aware maps were feasible on devices like the Nvidia Jetson Orin Nano. This approach enabled real-time object-aware map building while maintaining the advantages of VLMs' visual recognition performance.

In this study, we further refine our previous research, focusing on simplifying the map building and inference process to enable language driven object navigation on microcomputers. Our objective is to create a system where mobile robots can perform zero-shot object navigation based on human language instructions in mobile robots with limited resource. By separating computational process of VLMs and inference process, we aim to achieve both performance of VLMs and cost-efficiency of microcomputers. With this method, mobile robots can operate autonomously and effectively in diverse environments even with low-powered computers.

In this study, we introduce several new contributions that extend beyond our previous work:

- **Real-time zero-shot object navigation with minimal resources**: Our system is designed to navigate to objects based on human text instructions without the need for extensive computational resources or external internet connection. The system enhanced the autonomy of mobile robots in diverse environments.
- **Integration of VLMs with microcomputers**: We demonstrate the practical application of object-aware map using VLMs in microcomputers. It ensures cognitive visual understanding and text driven navigation can be achieved on hardware with limited processing power and memory.
- **Experimental validation on real-word environment**: We conduct experiments on microcomputers like Raspberry Pi, and evaluated the performance of our

optimized system in real-world scenario, to validate its efficiency and applicability in cognitive mobile robotics.

This paper is structured as follows: Section II reviews related work in the field, highlighting existing approaches and their limitations. Section III details our methodology, including the specific techniques employed to optimize VLMs for low-powered devices. In Section IV, we present our experimental results, demonstrating the system's performance on microcomputers such as the Raspberry Pi. Finally, Section V concludes the paper, discussing the significance of our method and exploring directions for further research.

II. Related Works

A. VLMs(Visual Language Models)

Visual Language Models (VLMs) presented a significant advancement in addressing vision-language problems. VLMs enabled systems to process both images and natural language prompts simultaneously. These models have been developed to tackle a variety of tasks, such as image-text retrieval, image captioning, and visual question answering[1]. Notable VLMs, including CLIP[2] and Flamingo[3], achieved this by encoding both text and images into a shared dimensional space, allowing for the processing of visual and textual information in the same vector space. With simply calculating the dot product between image and text embeddings, users can determine the most similar text prompt to a given image. By pre-training on billions of image-text pairs, these models can perform zero-shot image classification, achieving impressive results without requiring additional fine-tuning. This capability has opened new avenues for integrating VLMs into various applications where language-vision interaction is crucial in cognitive robotics.

B. Visual Language Mapping and Navigation

The use of VLMs has also expanded into the domains of mapping and navigation, where they are employed to enhance robots' ability to interpret and interact with their environment using human language. Several studies have explored the integration of VLMs into mapping systems[4,5] and navigation tasks[6,7]. These approaches enabled robots to understand natural language instructions and perceive objects with zero-shot recognition skills. For example, some studies have utilized visual language segmentation models[4], or combined RPN(region proposal network) and double VLM architectures with maximum ensemble methods[5], allowing robots to recognize objects using camera inputs. However, these VLM-based systems are computationally intensive, with millions of parameters that requires significant computing power to transform images and text into vector embeddings. The resulting map or object representations, consisting of high-dimensional vector embeddings, are often too large and complex to run efficiently on edge computing devices with limited memory resources.

C. DMAP(Query-based Object-aware Visual Language Mapping using Frame Division Method)

In response to the computational and storage challenges posed by existing VLM-based mapping systems, our recent research introduced DMAP, a more time-efficient and space-efficient approach to mapping with VLMs. Unlike other methods, such as VLMaps[4], which generate object-aware maps by calculating embeddings for every 2D-projected pixel using LSeg[8] which is heavy-weighted language segmentation model, DMAP[9] employs a set of optimization techniques to streamline the process. These techniques include keyframe selection, frame division, and a query-based method, all of which simplify the calculation process. This approach enabled the generation of vector-embedded maps on edge devices like the Nvidia Jetson Orin Nano, which would be incapable of running other conventional computationally demanding language mapping methods. For comparison, experimental results demonstrated that DMAP was 3.1 times faster and 8.9 times more space-efficient than other visual language mapping methods even on AI computers. However, despite these improvements, microcomputers such as the Raspberry Pi still struggle to run this module efficiently, indicating the need for further optimization for more low-powered devices.

This review of related works highlights the evolution of VLMs from general-purpose vision-language models to specialized systems for mapping and navigation in robotics. While significant progress has been made, the challenge of deploying these models on low-powered mobile robots remains a critical area for ongoing research, which this paper seeks to address.

III. Methodology

The proposed system enhances the DMAP framework by allowing mobile robots to utilize object-aware maps for navigation on resource-constrained platforms, such as microcomputers. While DMAP originally enabled real-time object aware map building and object navigation, the embedding process involved in mapping is computationally intensive. To address this challenge, our approach separates the embedding process from map building, enabling micro robots to leverage pre-built object-aware maps for efficient navigation. The methodology is divided into three key stages: (A) Data Acquisition, (B) Object-Aware Map Building, and (C) Inference. The system architecture is illustrated in Figure 1.

A. Data Acquisition Stage

In the Data Acquisition stage, the robot constructs an obstacle map using SLAM(Simultaneous Localization and Mapping). During this process, keyframes are captured and stored along with the robot's pose and observed LiDAR scan data, as depicted in Figure 1 (a). As the robot explores the environment, keyframe selection algorithm is employed to choose frames that provide the most informative content. This algorithm, based on our previous research, ensures that only the most valuable frames are selected for further processing. The keyframe selection process is governed by the following equations:

$$k_{t=kf}\colon f_t(\boldsymbol{P_t}) - f_{kf-1}\left(\boldsymbol{P_t} \cap \boldsymbol{P_{kf-1}}\right) > k_{th} \tag{1}$$

$$f_t(p) = w_o\left(d_{t,p}\right) = de^{\frac{d}{o}+1-\log o} \tag{2}$$

In this context, $d_{t,p}$ represents the distance between the camera and a specific point p in frame t. The variable o denotes the optimal distance between the camera and the point, which is the preferred distance for capturing the most informative data. The function $w_o\left(d_{t,p}\right)$ is a weighting function that reaches a maximum value of 1 when the distance $d_{t,p}$ is at the optimal distance o. This value decreases as the distance moves either closer to or further away from the optimal range. During keyframe selection, frames are saved if

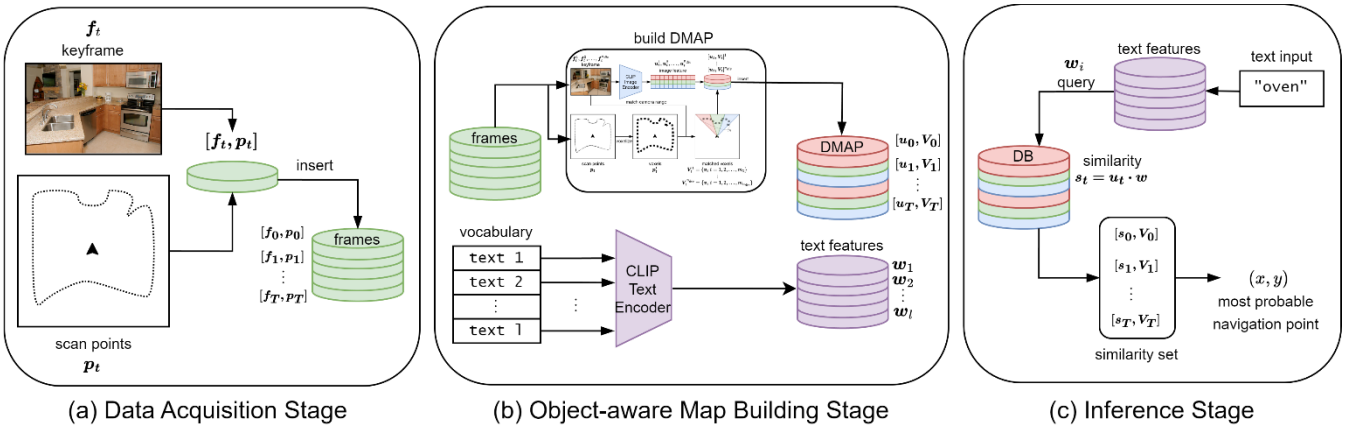


Fig. 1. Overview of proposed method. (a) data acquisition Stage of proposed method, (b) object-aware map building stage, (c) inference stage.

they exceed a predefined threshold k_{th} compared to previously selected keyframes, ensuring that only the most informative frames are retained for further processing.

B. *Object-Aware Map Building Stage*

In the Object-Aware Map Building stage, the selected keyframes from the Data Acquisition stage are processed on high-performance devices to construct object-aware maps, as shown in Figure 1 (b). These maps are then distributed to micro robots, enabling them to operate with object-awareness without the need for computationally intensive on-board processing.

The object-aware map includes both the obstacle map and the image embeddings of keyframes, along with observed voxels. During this stage, predefined text vocabularies are processed through a text encoder to generate text features that will be used in the inference stage. These vocabularies include lists of available objects in captured environment, semantic locations, and specific object descriptions (e.g., "dotted umbrella" or "a bottle containing black liquid"). By embedding both image and text features, the system equips micro robots with the ability to intelligently determine goal points based on text inputs during the inference stage.

C. *Inference Stage*

In the Inference stage, mobile robots utilize the pre-built map to localize themselves using a localization module derived from the SLAM map created in the Data Acquisition stage. The robots then accept human text inputs, which serve as instructions for navigating to specific objects or locations.

Upon receiving a text input, the robot searches for the goal position by calculating a probability map based on the input. For input instructions, the robot calculates similarities between the input text and the keyframes using the pre-computed embeddings. The system calculates voxel-wise probabilities by accumulating the similarities of captured voxels across all relevant frames which has selected with softmax similarities as shown in equation (3).

$$f_{t=rf}:\ ss_t = \frac{e^{s_t}}{\sum_j e^{s_j}} > ss_{th} \tag{3}$$

$$d_v = \sum_j d_{j,v},\ d_{j,v} = \begin{cases} 1, & v \in f_j \\ 0, & else \end{cases} \tag{4}$$

$$c_v = \frac{\sum_j ss_j d_{j,v}}{d_v} \tag{5}$$

$$p_v = c_v + \frac{d_v}{\max(d_j)} \tag{6}$$

These voxel-wise probabilities p_v, based on both average of softmax similarity c_v and observed frequency d_v, guide the robot in navigating to the desired object or location, ensuring efficient and accurate execution of the given instructions.

This methodology enables mobile robots with microcomputer to perform real-time, zero-shot object navigation using pre-built object-aware maps and text features, overcoming the computational limitations of resource-constrained systems.

IV. Experiments

This methodology was intended to optimize the computational efficiency of the system by separating encoding process of image embedding and text embedding during navigation. So, the experiment was focused on evaluating the system's accuracy using a pre-defined and pre-calculated text list corresponding to the available objects and locations in the environment.

A. *Experimental Setup*

Experiments were conducted using a Turtlebot4 mobile robot equipped with an OAK-D-PRO camera, an RPLIDAR-A1 2D LiDAR, and a Raspberry Pi 4B microcomputer with 4GB memory. The robot's localization was managed using the SLAM Toolbox[10], which employed 2D LiDAR-based Simultaneous Localization and Mapping (SLAM), while the Nav2[11] navigation framework, implemented in ROS2 (Robot Operating System), handled the navigation tasks.

During the experiment, the obstacle map was constructed using the SLAM Toolbox with odometry calculated from the 2D LiDAR. Subsequently, an object-aware map was created using the DMAP methodology, which combined information from both the camera and the 2D LiDAR before the inference stage. The object-aware map was built with 5,112 divisions derived from 1,704 keyframes. These keyframes were processed using ViT-B/16-SigLIP pre-trained weights from OpenCLIP[12], ensuring that the embeddings were optimized for the task at hand.

The available text list used in the experiment consisted of a comprehensive vocabulary set, including 17 object names like ‘umbrella’, ‘door’, ‘desktop’, 6 location names like ‘aisle’, ‘floor’, ‘wall’, and 15 specific descriptions of objects and locations like ‘shelf with books’, ‘a bag on a chair’. This predefined vocabulary allowed the robot to interpret human text instructions and navigate to the object based on the object-aware map. In this experiment, we can measure capability of our method to perform intelligent object navigation in resource-constrained system.

B. *Experiment Result*

In the experiments, the success rate was calculated based on the generated goal points. If the goal points was within 1-meter distance of the specified object's range, it’s considered success. Table 1 shows the success rates across different categories, including object names, locations, and descriptions of objects and locations. Figure 2 illustrates the paths generated by the navigation system toward the object goals, and in the right side, camera frame captured when the robot reached its goal to visualize whether it is valid.

TABLE I. EXPERIMENT RESULT IN TURTLEBOT4

Category	Texts	Success	SR
Object	17	11	65%
Location	6	5	83%
Description	15	9	60%
Total	38	25	66%

The results showed 66% of overall success rate with 65% for object names, 83% for location names, and 60% for descriptions of objects and locations. These results demonstrate that the system could generate paths and navigating to the correct object locations based on text inputs in microcomputers like Raspberry Pi with our method.

C. *Discussion*

The experimental results showed reasonable performance of proposed system with handling pre-calculated textual embeddings. The high success rates across different categories underscore the potential of our approach for deployment in real-world robotic applications, particularly in environments where computational resources are constrained.

One of the key advantages of this approach is its ability to function efficiently on microcomputers, such as the Raspberry Pi, which have limited processing power. By pre-calculating and distributing object-aware maps, the system circumvents the need for real-time, resource-intensive computations, enabling low-powered robots to perform complex navigation tasks based on natural language inputs.

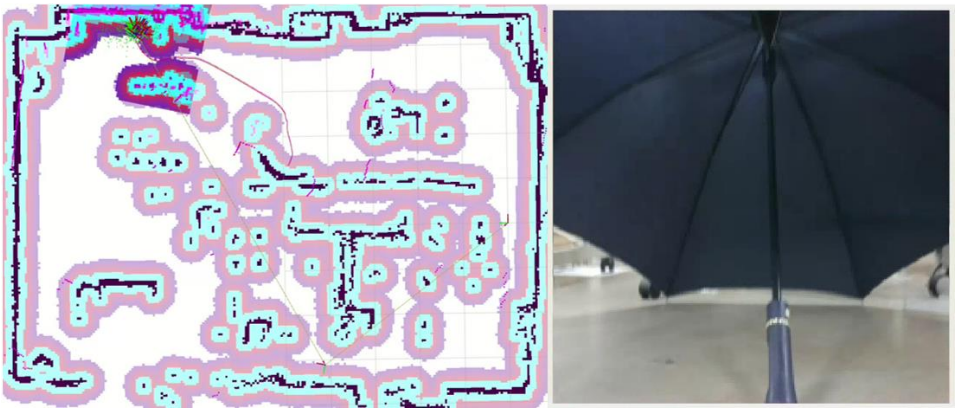
However, a limitation still exists that the system's reliance on pre-defined text lists. While the predefined vocabulary allowed for accurate navigation in the experimental setup, this approach may not be as flexible in more dynamic environments where objects and locations might not be covered by the predefined text. This limitation points to the need for further optimization, such as simplifying the tokenizer and text encoder, to allow the system to process a broader range of natural language inputs without being restricted to a fixed vocabulary.

In addition, while the system demonstrated strong performance in controlled conditions, future work could explore its robustness in more complex and unstructured environments. This includes handling ambiguous or imprecise text inputs, adapting to changes in the environment, and integrating real-time updates to the object-aware map.

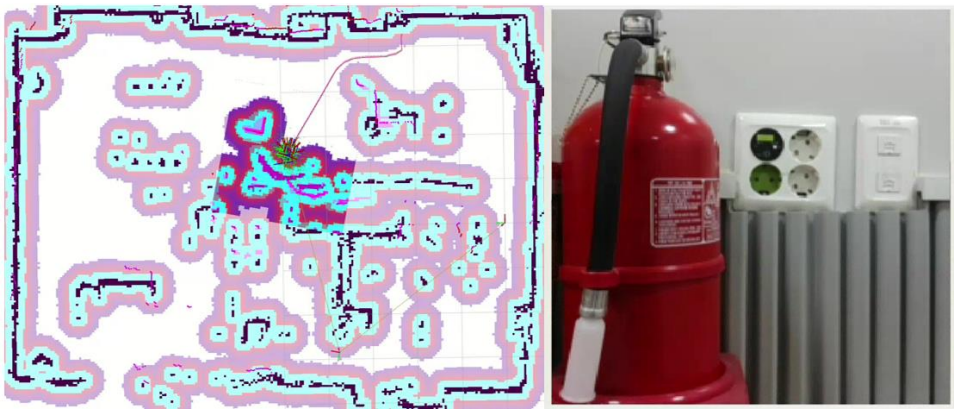
V. CONCLUSION

In this paper, we utilized DMAP[9], a lightweight object-aware mapping method, to enable robots equipped with microcomputers to perform object navigation based on natural language input. By distributing prebuilt maps and text features to microcomputers such as the Raspberry Pi, we demonstrated that even small, low-performance robots can execute zero-shot navigation tasks language input. The experiments demonstrated that the prebuilt DMAP system allowed mobile robots with microcomputers to navigate with object-aware intelligence, achieving an overall success rate of 66%, including a 60% success rate for descriptions of objects or locations.

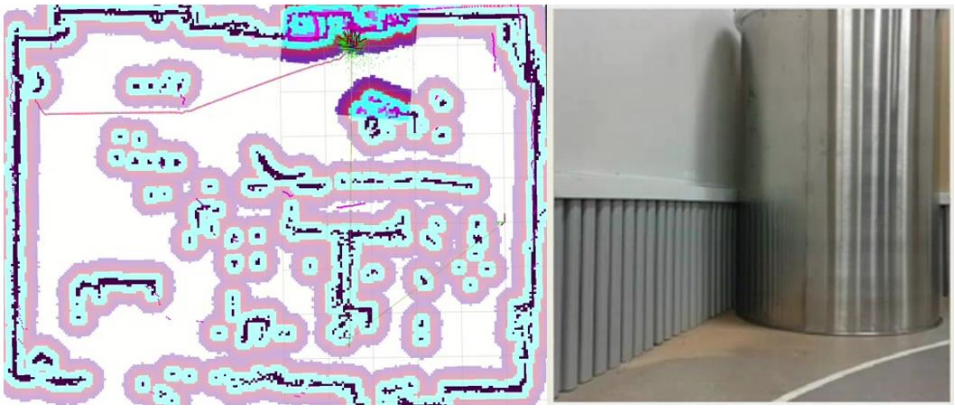
These results suggest that cognitive mobile robots equipped with microcomputers, despite their limited computing power, can effectively navigate and complete specific missions using object-aware maps. However, the system is currently limited by its reliance on predefined text lists, which restricts its ability to process every possible text input. Future research should focus on simplifying the tokenizer and text encoder, which would enable cognitive mobile robots to understand and navigate based on any natural language input, eliminating the need for predefined text lists and expanding their operational flexibility in dynamic environments.



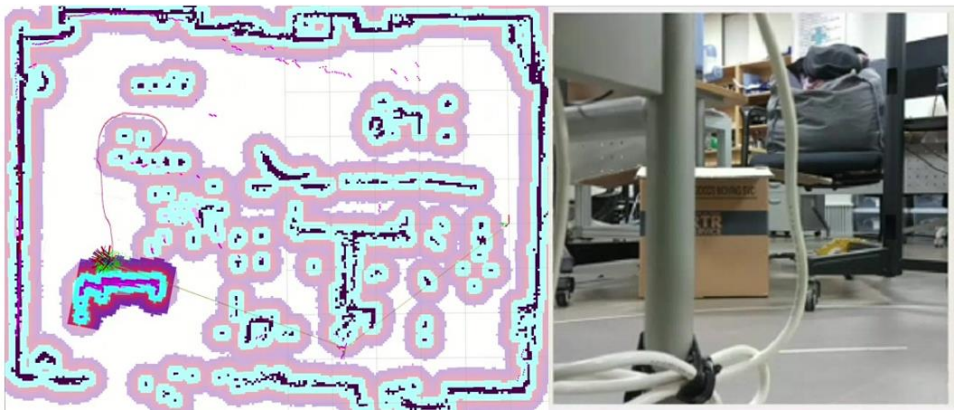
(a) umbrella



(b) fire extinguisher



(c) trash bin



(d) tangled wire

(e) green table

Fig. 2. Generated goal point, navigation path and frames after navigation ended. Goal point and navigation path is described as red line in front of robots. The camera frame was captured after navigation through path was ended and reached goal point.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (NRF-2022R1F1A1073799).

REFERENCES

[1] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, and J. Gao, “Vision-language pre-training: Basics, recent advances, and future trends,” *Foundations and Trends® in Computer Graphics and Vision*, vol 14, no. 3-4, pp. 163-352, 2022

[2] A. Radford, J.-W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, ..., and I. Sutskever, “Learning transferable visual models from natural language supervision,” *Proceedings of the 38th International Conference on Machine Learning (ICML)*, vol. 139, pp. 8748-8763, July, 2022.

[3] J. B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, …, and K. Simonyan, “Flamingo: a visual language model for few-shot learning,” *Advances in neural information processing systems (NeurIPS)*, vol. 35, pp. 23716-23736, 2022

[4] C. Huang, O. Mees, A. Zeng and W. Burgard, "Visual language maps for robot navigation," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608-10615, May, 2023.

[5] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo and D. Kappler, "Open-vocabulary queryable scene representations for real world planning," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11509-11522, May, 2023.

[6] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23171-23181, June, 2023.

[7] T. Guan, Y. Yang, H. Cheng, M. Lin, R. Kim, R. Madhivanan, ..., and D. Manocha, "LOC-ZSON: Language-driven Object-Centric Zero-Shot Object Retrieval and Navigation," *arXiv preprint arXiv:2405.05363*, 2024.

[8] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun and R. Ranftl, "Language-driven semantic segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[9] J. Yun, and P. Kim, “Query-based object-aware mapping for on-device visual language mapping and navigation,” unpublished.

[10] S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, 2783, 2021.

[11] S. Macenski, F. Martin, R. White, and J. Clavero, “The marathon 2: A navigation system,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.2718-2725, October, 2020

[12] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, ..., and J. Jitsev, "Reproducible Scaling Laws for Contrastive Language-Image Learning", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818-2829, June, 2023.