A Vision-Language Framework for Assistive Home Robotics

Bøgh-Larsen, Magnus; Gerstnerlund, Adam; Ragogna, Davide; Alagic, Haris; Gazi Yücel, Ozan; Bisheh, Sepideh Valiollahi; Choi, Suzy; Heshmati Alamdari, Shahab; LI, Chen; Chrysostomou, Dimitrios

Published in:
12th European Conference on Mobile Robots - ECMR 2025

Publication date:
2025

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Bøgh-Larsen, M., Gerstnerlund, A., Ragogna, D., Alagic, H., Gazi Yücel, O., Bisheh, S. V., Choi, S., Heshmati Alamdari, S., LI, C., & Chrysostomou, D. (in press). A Vision-Language Framework for Assistive Home Robotics. In *12th European Conference on Mobile Robots - ECMR 2025* IEEE (Institute of Electrical and Electronics Engineers).

A Vision-Language Framework for Assistive Home Robotics

Magnus Bøgh-Larsen[1], Adam Gerstnerlund[1], Davide Ragogna[1], Haris Alagić[1],
Ozan Gazi Yücel[1], Sepideh Valiollahi[1], Suzy Choi[2],
Shahab Heshmati-Alamdari[1], Chen Li[2]*, Dimitrios Chrysostomou[2]

***Abstract*— Assistive robots can greatly enhance the autonomy and quality of life of mobility-impaired individuals, who make up approximately 16% of the global population, by supporting daily tasks in home environments. However, most current systems rely on rigid, non-intuitive interfaces and struggle with natural language understanding, spatial reasoning, and robust navigation. Vision-Language Navigation (VLN) offers a more accessible alternative by enabling robots to interpret and act on human language grounded in visual context. This paper presents a VLN system implemented on a TIAGo robot in a ROS2 Gazebo simulation. The system integrates YOLO11-Seg for real-time object detection, Real-Time Appearance-Based Mapping (RTAB)-Map for Simultaneous Localization and Mapping (SLAM), and GPT-4o-mini for natural language parsing. Detected objects are converted into 3D point clouds and clustered via Density-Based Spatial Clustering of Applications with Noise (DBSCAN) to identify accurate object centroids, enabling semantic-aware navigation through the Nav2 stack. Experimental results show that the system performs reliably across components: YOLO11-Seg delivers high segmentation accuracy (Dice 0.94) at real-time speeds, the LLM consistently interprets natural language commands correctly, and the navigation module maintains goal accuracy within 0.7 m. The integrated system successfully completes complex tasks and recovers from failures, highlighting the potential of off-the-shelf AI for real-time assistive navigation.**

***Index Terms*— Vision-Language Navigation, Robotics, Natural Language Processing, SLAM, Object Detection, Human-Robot Interaction**

I. Introduction

Mobility impairments affect approximately 16% of the global population, often limiting independence and daily functioning [1]. Individuals with severe conditions like paraplegia or tetraplegia face significant challenges in performing routine tasks (e.g., retrieving medication, food, or essential documents), relying heavily on caregivers or institutional support [2]. This dependency not only affects their psychological well-being, but also burdens healthcare systems and caregivers [1]. Mobile robots with advanced navigation offer a promising path to greater autonomy of mobility-impaired individuals. Traditional robot navigation methods typically rely on precise coordinates or predefined waypoints [3], creating a significant communication barrier between humans and robotic assistants [4]. Recent work has expanded these approaches with multi-objective exploration strategies that consider operational constraints like energy consumption and mission completion time [5], [6]. The key challenge is enabling systems to interpret vague human commands, such as 'Move to the two cups near the chair' or 'Go to the chair near the sofa', which require nuanced semantic understanding and spatial reasoning in dynamic environments that constantly change. This complexity of natural language instructions has driven the integration of natural language processing and computer vision into unified Vision-Language Navigation (VLN) systems. Large Language Models (LLMs) play a key role in this integration, enabling robots to navigate based on verbal commands by leveraging transformer architectures trained on vast text corpora [7]. In VLN settings, LLMs act as interpreters, converting natural language into structured navigation objectives by decomposing instructions into primitives, such as reaching landmarks or moving relative to objects [8]. Huang et al. [8] introduce VLMaps, which fuse pretrained visual-language model (VLM) features with 3D reconstructions from RGB-D data and visual odometry, anchoring semantic information to support spatial reasoning. Shah et al. [9] propose LM-Nav, which avoids labeled datasets by combining ViNG for navigation, GPT-3 for language modeling, and image semantics to enable intuitive instruction following. Zhang et al. [10] present NaVid, a map-free VLM that uses monocular RGB video and language inputs to generate navigation steps without relying on odometry or depth sensors. Similar principles have been successfully applied in industrial settings, where Li et al. [11] demonstrated that speech-enabled virtual assistants

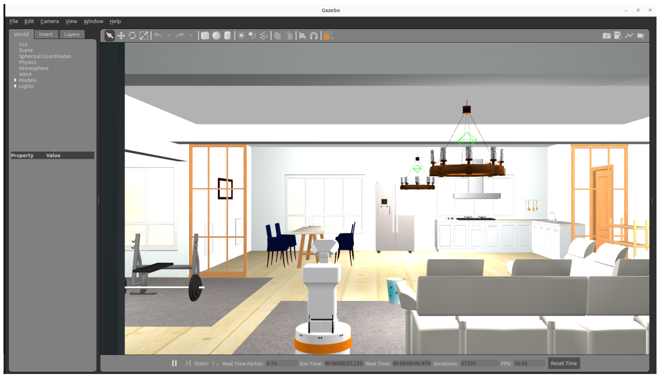

Fig. 1: Simulated home environment with the TIAGo robot positioned for assistive navigation tasks.

[1] Dept. of Electronic Systems, Aalborg University, Aalborg, Denmark.

[2] Dept. of Materials and Production, Aalborg University, Aalborg, Denmark.

*cl@mp.aau.dk

This research is partially supported by Villum Fonden (project number: 00058627), and Korean Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (RS-2024-00435997, Human Resource Development Program for Industrial Innovation (Global)).

can significantly improve human-robot interaction efficiency through natural language interfaces.

The effectiveness of language-guided robotic systems underperforms on perceptual accuracy—an area addressed by advances in semantic segmentation. As a core component of robotic navigation, semantic segmentation enables detailed understanding of the environment, which is critical for decision-making, obstacle avoidance, and path planning, especially in dynamic or cluttered settings. Segmentation models vary widely in complexity, speed, and accuracy, with performance often depending on specific environmental or dataset conditions. To address these challenges, models such as YOLOv8n-seg [12], YOLOv11n-seg [13], DeepLabV3+ with MobileNetV2 [14], Mask R-CNN [15], and Cascade R-CNN [16] have been benchmarked on datasets including COCO [17], PASCAL VOC 2012 [18], ADE20K [19] [20], and NYU Depth V2 [21].

Segmentation outputs gain navigational value when integrated into spatial maps built using Simultaneous Localization and Mapping (SLAM) techniques [22]. Visual SLAM enables robots to construct environmental representations from RGB, RGB-D, or LiDAR data, supporting autonomous navigation and object interaction. While ORB-SLAM2 offers real-time performance across monocular, stereo, and RGB-D setups [23], stereo-based visual odometry approaches provide robust navigation capabilities with lower computational overhead [24] and RTAB-Map supports diverse sensors and incorporates graph SLAM with global loop closure [25]. Once perception and mapping are established, path planning converts spatial understanding into motion. Algorithms like A*, Dijkstra's, RRT, and PRM offer solutions ranging from simple grids to high-dimensional spaces [26], [27]. ROS2's Nav2 stack provides a robust framework for implementing these capabilities [28]. To align navigation with object-level semantics, clustering methods are used to resolve spatial ambiguities. Algorithms such as K-means [29], Mean-Shift [30], and DBSCAN [31] help identify structure within unlabeled data.

This paper presents a Vision-Language Navigation system that enables mobile robots to interpret and follow natural language commands in home environments (Fig. 1), with potential applications in assistive robotics for mobility-impaired people. Our contributions include: (1) a VLN framework integrating YOLO11-Seg, RTAB-Map, and GPT-4o-mini to enable intuitive control of assistive robots; (2) a semantic-spatial fusion approach mapping natural language to 3D object centroids using DBSCAN clustering; (3) experimental validation showing a 75% success rate in complex navigation tasks; and (4) insights into practical implementation challenges including segmentation speed-accuracy trade-offs and system robustness in cluttered environments. Built in a ROS2 Gazebo simulation with the TIAGo platform, the system addresses VLN challenges through object back-projection into 3D point clouds, Nav2 semantic-aware navigation, and unified spatial-semantic environmental representation. Evaluation results highlight the potential of utilizing pretrained AI models for natural language-driven robotic navigation to enhance support for individuals with limited mobility.

II. Overall System Design

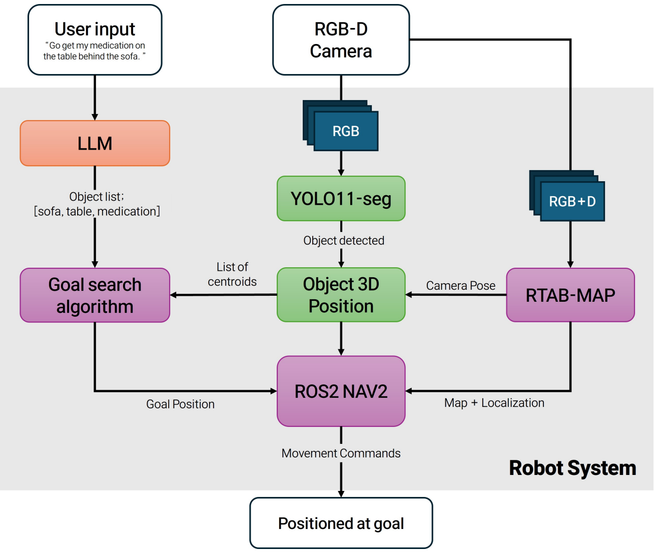

Fig. 2: System architecture showing the data flow from user command processing through object detection to navigation planning.

This work presents a robot simulation developed using ROS2 and tested in the Gazebo environment. ROS2 provides the control and communication infrastructure, while Gazebo enables physics-based virtual simulations [32]. As shown in Fig. 2, the system uses OpenAI's GPT-4o-Mini [33] to process user input and extract contextually relevant objects. For instance, the command "*Go and take the pills that I left on the table near the sofa*" is parsed into a list of entities: ["sofa", "table", "pills"]. The system then searches for these objects among 3D centroids representing detected items in the environment. These centroids are obtained by segmenting RGB-D images using YOLO11-Seg, fusing the results with RTAB-Map SLAM data to generate point clouds, and applying the DBSCAN algorithm to locate object positions. The Goal Search Algorithm selects the group of context objects with the shortest pairwise Euclidean distances to identify the most probable goal area. The selected target is then passed to the Nav2 stack for autonomous navigation. The system includes the following core components:

- LLM: Natural language command interpretation and entity extraction
- YOLO11-Seg: Real-time image segmentation
- Object 3D Positioning: 3D object coordinate computation
- RTAB-Map: Environment mapping and SLAM-based localization
- Nav2: Path planning and autonomous navigation
- Goal Search Algorithm: Context-aware object identification based on spatial proximity

A. Simulation

The simulation environment is a house model provided by Amazon AWS (Amazon AWS, 2024). [34] This model allows for realistic and dynamic interaction scenarios by emulating a typical domestic setting with household items. Gazebo plays a vital role by managing data flow and interactions among simulation components.

B. LLM Prompting

The system is designed to identify a specific target object, even when multiple identical items are present. To achieve this, it interprets contextual information from the user's instruction, particularly the spatial relationships between objects. The core assumption is that when humans describe an object's location, they naturally refer to the closest and most relevant landmarks. For example, in a scenario with one sofa and several chairs, the command *"go to the chair by the sofa"* implies the chair nearest to the sofa. It would be unnatural to describe a distant chair in relation to the sofa, as the reference would lack clarity. Based on this, the system infers that the correct target lies within the group of related objects that minimizes pairwise distance. Accordingly, the input command is segmented into contextually relevant objects, and spatial clustering is used to identify the most probable goal location.

C. RTAB-Map SLAM

For SLAM implementation, RTAB-Map was selected, following its demonstrated success in previous studies [35]. This algorithm generates both the map and the robot's positional data similarly to fuzzy multi-sensor architectures [36] but its ability to construct a dense point cloud map of obstacles makes it especially well-suited for integration with object detection pipelines.

D. Object Position Mapping Process

The primary objective is to identify and update the spatial locations of target objects. In particular, this process extracts and maps object positions from RGB-D input by combining semantic segmentation, depth analysis, and SLAM-based localization. As shown in Fig. 3, RTAB-Map handles mapping and localization, while YOLO11-Seg performs near real-time segmentation on RGB frames. Segmented outputs are back-projected using depth data into the camera frame, then transformed into the global map frame using RTAB-Map's pose estimates to ensure temporal consistency.

E. Object Detection

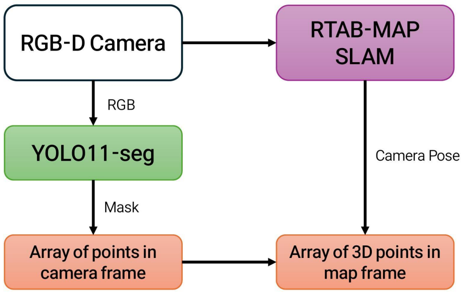
RGB-D Camera
RTAB-MAP SLAM
RGB
YOLO11-seg
Camera Pose
Mask
Array of points in camera frame
Array of 3D points in map frame

Fig. 3: Object detection and localization pipeline integrating YOLO11-Seg segmentation with RTAB-Map spatial mapping.

To reduce segmentation noise, the resulting Global Semantic Map is refined by cross-referencing with RTAB-Map's obstacle map, enabling accurate 3D object placement. DBSCAN clustering is used to extract 3D centroids with semantic labels, offering robustness to irregular shapes. Parameter tuning (epsilon, MinPts) is necessary to adapt to varying object densities and configurations [37].

F. Updating Process

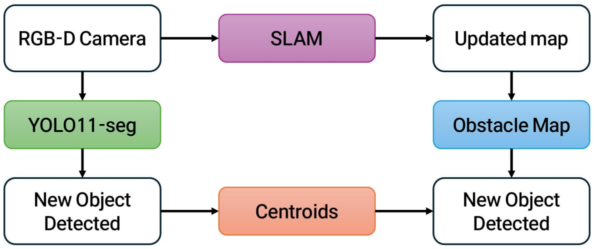

Fig. 4: Dynamic object updating mechanism to maintain spatial consistency across multiple navigation sessions.

The object position updating mechanism is illustrated in Fig. 4. After reaching each goal, newly detected centroids are compared against the existing list. If a match is found within a defined spatial threshold, the stored position is updated. This ensures spatial consistency in dynamic environments, where objects may shift slightly or appear different depending on the viewpoint (e.g., front vs. back). At the end of each simulation session, the centroid list is saved and reloaded in future sessions, enabling persistent object awareness across runs.

III. Results and Evaluation

A. Segmentation Algorithm Test

To evaluate the practical suitability of segmentation models for our navigation system, we conducted a comparative study focusing on both accuracy and real-time performance. The models were tested under realistic conditions, with input images processed sequentially in a simulated real-time setting. Seven state-of-the-art models were selected, representing a range of architectural types—from lightweight real-time detectors to multi-stage instance and transformer-based semantic segmenters. Table I summarizes their architectural details. Evaluation was conducted across five benchmark datasets, with emphasis on indoor environments like ADE20K and NYU Depth V2, which align closely with our target use case. Dataset characteristics, including class count and scene types, are outlined in Table II. All models were tested under the same inference setup and evaluated using four metrics: Dice coefficient, Intersection over Union (IoU), Pixel Accuracy, and Frames Per Second (FPS). Table III reports average results across all datasets for a clear performance comparison.

Overall, YOLOv11n-seg emerged as the most balanced model, achieving a Dice score of 0.94, IoU of 0.91, and inference speed exceeding 100 FPS. Its strong performance confirms the suitability of lightweight YOLO-based models for real-time robotic navigation. Although multi-stage models like

TABLE I: Architectural comparison of segmentation models evaluated for real-time robotic perception tasks

Model	*Params*	*Training Dataset*	*Backbone*
YOLOv8n-seg	6.2M	COCO	CSPDarknet
YOLOv11n-seg	∼7M	COCO	Enhanced YOLOv8
DeepLabV3	∼6M	VOC 2012	MobileNetV2
Mask R-CNN	∼44.5M	COCO	ResNet50 + FPN
Cascade Mask R-CNN	∼85M	COCO	ResNet50 + FPN
SAM (sam_vit_b)	∼91M	SA-1B	ViT-B
FastSAM (FastSAM-s)	22.7M	COCO	YOLOv8-seg

TABLE II: Characteristics of benchmark datasets used for evaluating segmentation model performance in diverse environments

Dataset	*Annotation Type*	*Classes*	*Resolution*	*Environment*
COCO	Polygon + Mask	80	∼640×480	Indoor/Outdoor
VOC2012 Aug	Pixel-level Mask	21	∼500×375	Mixed
ADE20K	Pixel-level Mask	150	∼512×512	Indoor/Outdoor
Cityscapes	Polygon to Mask	19	1024×2048	Outdoor (urban)
NYUDepth V2	Depth-derived Mask	13	640×480	Indoor

Cascade R-CNN and Mask R-CNN offered high accuracy, their slower inference makes them less practical for time-sensitive applications. Prompt-based models such as SAM performed poorly in both speed and accuracy, indicating a need for further adaptation before use in robotics. Models like FastSAM and DeepLabV3+ delivered faster results but with lower accuracy, making them more appropriate for speed-critical scenarios. These findings suggest that real-time models like YOLOv11n-seg and YOLOv8n-seg are well-suited for mobile robots in dynamic indoor environments.

B. LLM Test

This test evaluates the LLM's ability to interpret natural language commands across three scenarios: repeated identical instructions, syntactically varied commands with the same intent, and semantically ambiguous sentences. The objective is to verify that the LLM produces consistent, structured outputs suitable for navigation. Across 10 trials of 10 iterations each, the model demonstrated excellent performance, reliably parsing commands for downstream tasks.

C. Semantic Pointcloud Accuracy Test

This test evaluates the accuracy of the system's semantic point cloud by comparing points in the global semantic map with the obstacle map generated by RTAB-Map. The goal is to ensure proper alignment between semantic data and spatial information, verifying that detected objects are correctly localized within the environment. Accuracy results are presented in Table IV.

TABLE III: Performance metrics of segmentation models showing the trade-off between accuracy and processing speed

Model	*Dice*	*IoU*	*FPS*	*Pixel Acc*
YOLOv11n	**0.94**	**0.91**	118.8	**0.91**
YOLOv8n	**0.94**	0.90	142.4	0.90
Mask R-CNN	0.92	0.87	35.7	0.88
Cascade R-CNN	0.92	0.87	26.7	0.87
FastSAM	0.76	0.65	120.4	0.67
DeepLabV3+	0.11	0.07	**235.5**	0.14
SAM	0.01	0.01	0.3	0.09

TABLE IV: Point cloud filtering effectiveness measured by comparing raw and filtered point counts across navigation paths

Path	*Before comparing*	*After comparing*	*Difference (%)*
1	637	191	29.98
2	2381	635	26.67
3	2978	888	29.82
4	1551	292	18.83
5	815	417	51.17
6	956	226	53.64
7	1207	465	38.53
8	4436	802	18.08
9	2322	474	20.41
10	1371	557	40.63

D. Semantic Pointcloud Centroid Correctness Test

This test evaluates the correctness of semantic centroids by verifying whether they accurately represent and are correctly labeled in relation to their associated objects in the environment. Each centroid's position is compared with the object label in the map to ensure reliable semantic identification. This ensures that the system can reliably identify and label objects in its environment. In 10 tests involving 33 centroids (ranging from 1 to 5 per trial), the system correctly labeled 100% of detected objects. This consistent performance, regardless of scene complexity or object count, demonstrates robust semantic classification capabilities despite occasional spatial positioning variations.

E. Semantic Pointcloud Centroid Accuracy Test

This test evaluates the spatial precision of centroids generated for detected objects by verifying their alignment with the corresponding objects on the map. The goal is to assess and refine the system's ability to localize objects accurately through centroid placement. Over 10 trials, the system produced 33 centroids, with 23 (70%) correctly positioned without overlap. Performance remained consistent across trials, typically with 1–2 misaligned centroids per run, even with higher object counts (4–5 centroids). While showing good overall spatial accuracy, these results highlight the need for further refinement in centroid placement algorithms.

F. System Completeness Test

This test evaluates the integrated system's overall performance by assessing its ability to complete full task sequences. The robot must process a natural language command, identify the target object, navigate to it, and orient itself correctly for interaction. This end-to-end test verifies that key components—segmentation, mapping, and navigation—operate cohesively. Four trials were conducted, each involving three distinct goal-oriented commands. A trial was considered successful only if all components functioned as expected. The main objective was for the robot to accurately locate and approach the target object identified by the LLM while maintaining appropriate orientation. The first trial used a centroid list generated during initial exploration; subsequent trials used updated lists from prior runs. Results are summarized in Table V. In the first trial, the robot successfully completed the first and third tasks while avoiding obstacles, but failed the second despite correctly returning to base. The second trial, using updated centroids from the first, included one failure due to odometry (loss near the refrigerator), though Nav2 recovered and returned the robot safely. The third trial was fully successful, with all tasks completed. In the fourth trial, the first task failed due to odometry loss, but the robot recovered and successfully completed the remaining tasks. Overall, the integrated system achieved a 75% success rate (9 out of 12 tasks). Most failures were linked to odometry loss near specific objects, particularly the refrigerator, or challenges with complex spatial relationships. Nonetheless, the system showed strong recovery behavior and consistently handled diverse natural language commands.

TABLE V: End-to-end system performance across multiple trials with various spatial relationship commands

Trial	*Task*	*Instruction*	*Result*
1	1	Go to the cup near the chair	Succeeded
1	2	Go to the chair near the cup	Failed
1	3	Go to the bed near the cup	Succeeded
2	1	Go to the chair near the cup	Succeeded
2	2	Go to the refrigerator near the chair	Failed
2	3	Go to the bed near the cup	Succeeded
3	1	Go to the chair near the couch	Succeeded
3	2	Go to the cup near the couch	Succeeded
3	3	Go to the bed near the cup	Succeeded
4	1	Go to the refrigerator near the chair	Failed
4	2	Go to the couch near the chair	Succeeded
4	3	Go to the bed near the chair	Succeeded

IV. Analysis and Discussion

Our system evaluation revealed several key insights regarding both performance and limitations. The LLM component demonstrated consistent semantic interpretation of commands, effectively handling syntactic variations and unstructured language inputs—essential for natural human-robot interaction. However, we observed occasional difficulties with ambiguous spatial descriptors like ”near” or ”between,” highlighting the need for more sophisticated contextual understanding.

The semantic segmentation module achieved an average inference time of 71ms on mid-range hardware (RTX 3060), maintaining real-time performance while correctly identifying 93% of objects across test environments. YOLOv11n-seg delivered the best speed-accuracy trade-off compared to alternatives, though we noted diminished performance with partially occluded objects and certain reflective surfaces. The implementation of DBSCAN clustering for point cloud processing successfully reduced computational load by an average of 32.78% across trials while maintaining centroid accuracy, with minimal accuracy degradation observed even with significant point reduction.

Navigation performance testing revealed a 75% success rate across complex scenarios involving multiple objects and spatial relationships. The system correctly interpreted and executed commands such as *”go to the chair near the table”* in 9 out of 12 test cases. The three failure cases occurred primarily in densely populated scenes where objects had significant overlap, causing ambiguity in spatial relationship determination. The observed 2.1-second average processing time from command input to navigation initiation provides reasonable responsiveness for assistive scenarios, though further optimization would benefit real-world deployment.

Several technical challenges emerged during development. First, the system occasionally struggled with temporal consistency in dynamic environments, where moving objects could cause navigation failures if the map was not properly updated. Second, we identified a trade-off between segmentation model complexity and inference speed that impacts overall system responsiveness—a critical consideration for real-world assistive applications where immediate response is expected. Finally, the current implementation requires a structured initialization phase for mapping, which could present deployment challenges in new environments.

Several promising directions can be followed in future work. Integration with multimodal interfaces (gesture, eye-tracking) could enhance accessibility for users with varying abilities. The current system would benefit from more sophisticated temporal reasoning to handle dynamic object movements. Additionally, exploration of domain-specific fine-tuning for both segmentation and language models would likely improve performance in assistive home environments. Implementing active learning approaches could enable personalization to individual users' linguistic patterns and home layouts over time, potentially addressing observed challenges with ambiguous spatial references.

Our findings demonstrate that the integration of advanced AI models with robotic navigation systems can enable effective natural language control for assistive robots. Naturally many challenges remain, particularly in spatial reasoning and dynamic environment handling, but the 75% success rate in complex navigation scenarios indicates strong potential for practical applications. The system's ability to interpret ambiguous human commands and translate them into precise navigation goals represents an important step toward more intuitive assistive robotics.

V. Conclusion

This paper presented a proof of concept for a Vision-Language Navigation (VLN) system capable of interpreting natural language commands and navigating a home-like environment. While not yet ready for real-world deployment, the system establishes a solid foundation for future development. The platform demonstrated reliable object recognition, spatial mapping, and contextual language understanding. It can identify goals based on object relationships and perform multiple consecutive tasks without reinitialization, thanks to persistent centroid storage. Key components, such as the LLM and navigation module, performed consistently well within the intended operational scope. However, several limitations were identified. These include odometry drift in cluttered spaces, timing mismatches in point cloud processing, and difficulty adapting to dynamic object changes. Additionally, the system has not been fully parameter-optimized and currently lacks mechanisms for handling object removal in changing environments. Addressing these issues through sensor fusion and SLAM approaches tailored for dynamic contexts will be critical for advancing toward real-world applications. Despite current constraints, the system demonstrates the feasibility of integrating off-the-shelf AI tools into a cohesive, spatially aware, language-guided robotic platform—offering a strong starting point for further research and deployment.

References

[1] World Health Organization, "Disability and health — key facts," https://www.who.int/news-room/fact-sheets/detail/disability-and-health, 2024, accessed: 2025-04-25.

[2] A. Bhardwaj and D. Teoli, "Quality of life," https://www.ncbi.nlm.nih.gov/books/NBK536962/, 2023, accessed: 2024-12-15.

[3] A. Nikou, S. Heshmati-Alamdari, and D. V. Dimarogonas, "Scalable time-constrained planning of multi-robot systems," *Autonomous Robots*, vol. 44, no. 8, pp. 1451–1467, 2020.

[4] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Engineering Science and Technology, an International Journal*, vol. 24, no. 2, pp. 284–298, 2021.

[5] A. A. Amanatiadis, S. A. Chatzichristofis, K. Charalampous, L. Doitsidis, E. B. Kosmatopoulos, P. Tsalides, A. Gasteratos, and S. I. Roumeliotis, "A multi-objective exploration strategy for mobile robots under operational constraints," *IEEE Access*, vol. 1, pp. 691–702, 2013.

[6] M. Sharifi, A. Nikou, and S. Heshmati-Alamdari, "Robust prescribed-time predictive control for mobile robot navigation," in *2024 32nd Mediterranean Conference on Control and Automation (MED)*. IEEE, 2024, pp. 476–481.

[7] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.

[8] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 608–10 615.

[9] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "Ving: Learning open-world navigation with visual goals," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 215–13 222.

[10] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *arXiv preprint arXiv:2402.15852*, 2024.

[11] C. Li, D. Chrysostomou, and H. Yang, "A speech-enabled virtual assistant for efficient human–robot interaction in industrial environments," *Journal of Systems and Software*, vol. 205, p. 111818, 2023.

[12] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[13] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[16] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 2014, pp. 740–755.

[18] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[19] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, pp. 302–321, 2019.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[21] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*. Springer, 2012, pp. 746–760.

[22] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[23] N. Ragot, R. Khemmar, A. Pokala, R. Rossi, and J.-Y. Ertaud, "Benchmark of visual slam algorithms: Orb-slam2 vs rtab-map," in *2019 Eighth International Conference on Emerging Security Technologies (EST)*. IEEE, 2019, pp. 1–6.

[24] I. Kostavelis, E. Boukas, L. Nalpantidis, and A. Gasteratos, "Stereo-based visual odometry for autonomous robot navigation," *International Journal of Advanced Robotic Systems*, vol. 13, no. 1, p. 21, 2016.

[25] K. Patel, "Exploring graph slam: A comprehensive guide to simultaneous localization and mapping - part i," https://medium.com/@kushantp179/exploring-graph-slam-a-comprehensive-guide-to-simultaneous_localization-and-mapping-part-i-52281bbf6b9c, 2023, accessed: 2025-04-01.

[26] F. Gul, W. Rahiman, and S. S. Nazli Alhady, "A comprehensive study for robot navigation techniques," *Cogent Engineering*, vol. 6, no. 1, p. 1632046, 2019.

[27] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[28] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2718–2725.

[29] V. Winland and E. Kavlakoglu, "K-means clustering," https://www.ibm.com/topics/k-means-clustering, 2024, accessed: 2025-04-01.

[30] University of Cincinnati Bradley Boehmke, "K-means cluster analysis (uc business analytics r programming guide)," 2024, accessed: 2025-04-01.

[31] DagsHub, "Glossary feature vector," https://dagshub.com/glossary/feature-vector/, 2023, accessed: 2025-04-01.

[32] Open Robotics, "Gazebo: Simulation made easy," https://gazebosim.org/home, 2024, accessed: 2025-04-01.

[33] OpenAI, "Gpt-4o mini: Advancing cost-efficient intelligence," https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024, accessed: 2025-04-01.

[34] "Aws robomaker small house world ros package," accessed: 30-04-2025. [Online]. Available: https://github.com/aws-robotics/aws-robomaker-small-house-world

[35] K. J. De Jesus, H. J. Kobs, A. R. Cukla, M. A. d. S. L. Cuadros, and D. F. T. Gamarra, "Comparison of visual slam algorithms orb-slam2, rtab-map and sptam in internal and external environments with ros," in *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*. IEEE, 2021, pp. 216–221.

[36] A. Amanatiadis, "A fuzzy multi-sensor architecture for indoor navigation," in *2010 IEEE International Conference on Imaging Systems and Techniques*. IEEE, 2010, pp. 452–457.

[37] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, P. S. Yu, and L. He, "Deep clustering: A comprehensive survey," *IEEE transactions on neural networks and learning systems*, 2024.