Intelligent LiDAR Navigation: Leveraging External Information with Semantic Maps–LLM as an Actual Copilot

A project of the 2024 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University

https://robotics.shanghaitech.edu.cn/teaching/robotics2024

Fujing Xie, Xinzhe Liu
{xiefj, liuxzh12023}@shanghaitech.edu.cn

April 2024

Abstract. Traditional robot navigation has primarily relied on occupancy grid maps and laser-based sensing, as exemplified by the widely-used move_base ROS package. Unlike robots, humans navigate using not only spatial maps and physical distances but also incorporate external information such as elevator maintenance notifications from emails or experiential knowledge like the need for special access through certain doors. With the development of Large Language Models (LLMs), which can comprehend and process textual information, there is now an opportunity to infuse robot navigation systems with human-like understanding. In this project, we propose using osmAG(Area Graph in OpensStreetMap textual format), an innovative semantic topological hierarchical map representation to bridge the gap between move_base capabilities and contextual interpretation ability of LLMs. Our approach facilitates a more intelligent approach to robot navigation that leverages a broader range of informational inputs and yet still remain the robust capabilities of traditional robotic navigation.

1. Introduction

Recently, Large Language Models (LLMs) have demonstrated great potential in robotic applications by providing essential general knowledge for situations that can not be pre-programmed beforehand. Generally speaking, mobile robots need to understand maps to execute tasks such as localization or navigation. In contrast to commonly used map representations, such as occupancy grid maps or point clouds, osmAG (Area Graph in OpensStreetMap format) is stored in a XML textual format naturally readable by LLMs. Furthermore, conventional robotic algorithms such as localization and path planning are compatible with osmAG, facilitating this map representation comprehensible by LLMs, traditional robotic algorithms and humans.

Motivated by those, our project involves the integration of osmAG(Area Graph in OpensStreetMap format)[1] with ROS(Robot Operating System) to replace the traditional grid map used in move_base to implement a move intelligent path planning. In this project, We used a LLM (large language model) –ChatGPT as copilot to understand exteral information like emails and the semantic map to make navigation more intelligent. First given starting point and destination, path planning will be done by osmAG planning return areas and passages for robot to go through, then the planning results will be confirmed or rejected by LLM which aware of current situation from previous information like emails. After confirmed by LLM, the path would be send to ROS move_base to execute. The actual navigation is been done by move_base within a grid map rendered by osmAG with a LiDAR which is a robust and well-developed ros package. And during navigation, the system would document doors are successfully passed by robot or not, since not all doors are opened in a building all the time, this document would be considered as experience for navigation next time. Path planning algorithms should be intelligent to change with respect to environment change, for example:

1. A robot should avoid a crowded lobby during a graduation party.
2. A robot should avoid a elevator during its maintenance.

2. State of the Art

2.1. ROS Navigation and move_base

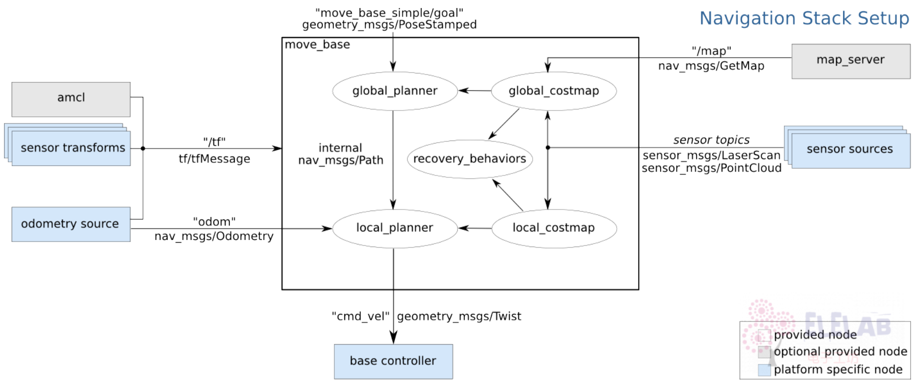

Figure 1. ROS navigation workflow

ROS (Robot Operating System) navigation is a powerful framework within the ROS ecosystem designed to facilitate autonomous robot navigation in various environments. It provides a collection of packages and tools that enable robots to perceive their surroundings, plan safe paths, and navigate autonomously. The workflow of ROS navigation is shown in Fig 1.

One of the key components of ROS navigation is the move_base package. move_base is a ROS node that serves as the core of the navigation stack. Its primary function is to take in the robot's current position, a goal position, and other relevant sensor information, then plan a collision-free path to navigate the robot from its current location to the goal location.

Here's how move_base typically works:

1. Global Planner: move_base utilizes a global planner to compute a high-level path from the robot's current position to the goal position. This path is generated based on a global costmap, which represents the environment's static obstacles.

2. Local Planner: Along the way, move_base employs a local planner to handle real-time obstacle avoidance and fine-tune the robot's trajectory to navigate around dynamic obstacles. The local planner operates within a local costmap, which provides information about obstacles in the robot's immediate vicinity.
3. Sensor Integration: move_base integrates data from various sensors such as laser scanners, depth cameras, or odometry to accurately perceive the environment and update the costmaps.
4. Feedback and Control: Throughout the navigation process, move_base provides feedback to the higher-level systems and controllers, adjusting the robot's velocity commands based on sensor inputs and path tracking errors.
5. Behavior Configuration: move_base offers extensive configuration options to tailor its behavior to different robot platforms and navigation scenarios. This includes parameters for obstacle avoidance strategies, motion constraints, recovery behaviors, and more.

Overall, move_base plays a critical role in enabling ROS-powered robots to navigate autonomously in complex and dynamic environments, making it a fundamental component for robotics applications ranging from mobile robots to autonomous vehicles.

2.2. RViz

RViz is essentially a 3D visualizer for displaying sensor data and state information from ROS. It can show data from a wide array of sources, including cameras, lasers, depth sensors, and more, all in real time. This capability makes it an invaluable tool for robotics developers who need to understand how their robots are interacting with their environments or for those who wish to simulate how a robot would perceive its surroundings.

The tool's interface is divided into displays and panels. Displays are used for visualizing different types of data, such as 2D and 3D data, including laser scans, images, and point clouds. Panels, on the other hand, provide a way to interact with this data, offering controls for viewing, managing the displays, and configuring the visualization settings. This arrangement allows for a highly interactive and flexible visualization environment that can be tailored to the needs of any ROS project.

RViz is not just for passive visualization; it can actively interact with ROS messages. For example, it can subscribe to topics, display sensor data as it's being published, and even publish messages to topics, allowing users to interact with a robot or simulation in real-time. This makes RViz an essential tool for tasks such as debugging complex robotics algorithms or planning and monitoring robot trajectories in both simulated and real-world applications.

In our project, we will use RViz to visual Area Graph, LLM global path, move_base local path, and use it to interact with our software by specified start area or goal area.

2.3. LLM Guided Robotics

The integration of robots with natural language models to enhance human interaction and decision-making in robotics has been studied in recent research. PaLM-E[2] is an innovative multimodal language model that integrates real-world sensor data, like images and robot state estimations, into language models, allowing for more grounded and effective decision-making in robotics and multimodal reasoning tasks. [3] fine-tunes PaLM-E[2] with robotic trajectory data to output robot actions. [4] uses LLM to help with object rearrangement. [5] utilizes LLM's ability to generate code to resolve a task and generates step for robots to execute, presenting a novel approach in robotics automation.[6] utilizes LLMs to build a semantic cost map that guides a motion planner to produce trajectories that meet motion constraints provided by a user in text format. [7] explores using LLM to encode human's high-level semantic instruction and output actionable trajectories using a transformer decoder. [8] is a visual-language navigation framework that enables robot agents to follow human navigation instructions and to generate route descriptions to humans.

[9] introduces Autonomous GIS as a revolutionary approach in geographic information systems (GIS) leveraging Artificial Intelligence (AI), specifically Large Language Models (LLMs) like GPT-4, to automate spatial data collection, analysis, and visualization. Autonomous GIS, as conceptualized, stands on the brink of transforming how spatial problems are addressed, making GIS technologies more intuitive and accessible to a broader audience beyond the confines of expert users. Central to the Autonomous GIS framework is its AI-powered autonomy, aiming for self-generating, self-organizing, self-verifying, self-executing, and self-growing capabilities. The authors present a prototype system, LLM-Geo, which encapsulates these ambitions by demonstrating the ability to take on spatial questions and autonomously navigate through the necessary geoprocessing steps to yield accurate results. This is achieved through a blend of natural language understanding, reasoning, and coding prowess inherent in LLMs. LLM-Geo's proficiency was validated through various case studies, showing its potential to accurately perform tasks like population analysis near hazardous sites, visualizing human mobility trends, and COVID-19 impact assessments with minimal human input. This paper heralds a paradigm shift in GIS, from traditional, heavily manual systems to AI-driven, autonomous systems. The envisioned Autonomous GIS aims not only to automate routine tasks but also to unlock new possibilities in spatial analysis by making it more efficient, reducing errors, and enabling complex task handling that would be challenging or impossible for humans alone. This transition towards an AI-powered GIS is seen as a crucial step towards democratizing GIS technologies, making spatial analysis more accessible, and potentially fostering innovation across a wide array of applications from urban planning and environmental monitoring to disaster management and public health. Our project also try to utilize LLM to handle GIS information related to robotics, this studies collectively highlight the versatility of LLMs in enhancing map or GIS (Geographic Information System) comprehension, which could benefit future robotics applications.

2.4. LLM in path planning

[10] explores an innovative approach to task planning in robotics using scene graphs and large language models (LLMs). It addresses the challenge of interpreting natural language tasks in robotics, leveraging recent advancements in semantic, metric, and topological mapping to enhance the understanding and execution of complex

tasks by autonomous robots. The primary contribution of the paper is the integration of LLMs with hierarchical metric-semantic models to translate natural language tasks into Linear Temporal Logic (LTL) formulas. This translation enables optimal hierarchical LTL planning over scene graphs, which represent the environment in a structured format comprising nodes and edges linked with semantic labels. The authors develop a hierarchical planning domain that encapsulates the attributes and connectivity of the scene graph and the task automaton. They also propose an LLM heuristic function to provide semantic guidance, aiming to enhance planning efficiency. The environment is depicted as a scene graph, capturing various elements like rooms, objects, and occupancy in a unified hierarchical format. This graph facilitates the grounding of semantic concepts from natural language instructions into the physical elements of the scene. LLMs, such as GPT-3 and BERT, are utilized to translate the structured representation of tasks into LTL automata. This step is crucial for linking natural language commands to specific semantic and metric elements within the scene graph. The planning system employs a hierarchical approach, where the tasks are broken down into sub-tasks at different levels of the scene graph. This structure allows for detailed and efficient planning, considering both high-level objectives and low-level operational constraints. The authors introduce a novel heuristic function based on LTL that complements the LLM heuristic. This multi-heuristic approach aims to balance the admissibility and consistency of the planning process, ensuring both optimality and efficiency.

2.5. osmAG

The osmAG (Area Graph in OpensStreetMap format) is stored in a XML textual format naturally readable by LLMs.

1. osmAG is a 3D hierarchical, topometric and semantic map representation that just stores the most important aspects of an environment crucial for robotic algorithms.
2. osmAG is stored in OpenStreetMap XML textual format, which is naturally readable by LLMs.
3. osmAG is a very compact representation that LLMs with limited token number can benefit from.
4. osmAG only stores permanent structures, so it is stable over time.
5. osmAG is easy to acquire using 3D point clouds, 2D occupancy grid maps or CAD files. Possible generation through CAD files enables robots to process prior information of places that itself or other robots have not visited beforehand.
6. Conventional robotic algorithms such as localization and path planning based on osmAG have been developed, so LLMs's behavior is easy to be monitored and verified by traditional robotic algorithms, e.g. to detect and prevent dangerous behavior.
7. osmAG can be easily visualized by JOSM (Java OpenStreetMap Editor) and in ROS's rviz, which makes it easy for humans to visualize and edit the map, which bridges the gap between map visualization and map representation. This way humans can interact with LLMs through the map.

   One example of osmAG is shown in Fig ??.

3. System Description

An overview of the pipeline is illustrated in Fig. 3 There are several element of the

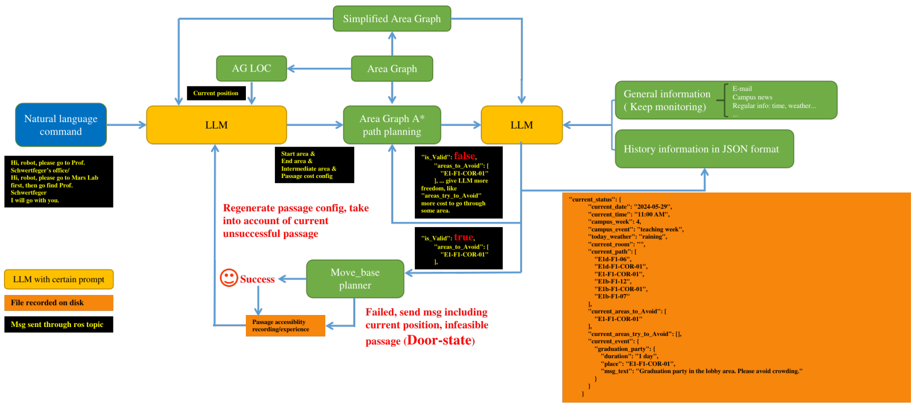

Figure 2. An overview of entire pipeline

entire pipeline:

1. Natural Language Command: The process starts with a user issuing a natural language path planning command.
2. LLM Processing: This command is interpreted by a large language model, which is responsible for understanding the command and translating it into 'start area' and 'end area' to send to a traditional A* algorithm on osmAG map. Also, this part takes previous saved experience as consideration to give cost to each passage.
3. Area Graph: semantic map representation that only capture permanent structure like walls or doors.
4. Simplified Area Graph: The area graph is simplified, delete unmecessary information in order to send to a LLM.
5. General information: information such as current time, weather, campus news that could effect path planning task is given to LLM.
6. Area Graph A Path Planning*: Using the A* path planning algorithm, the system attempts to find an optimal path from a start area to an end area using osmAG. The output of this path planning algorithm is a sequence of room names, for example: ["E1d-F1-06", "E1d-F1-COR-01", "E1-F1-COR-01", "outside", "E1b-F1-COR-01", "E1b-F1-07"]
7. LLM path evaluation: LLM need to check traditional planned path is valid or not according to current information, for example during elevator maintenance, a path going through elevator is invalid. If the path is valid ("is_Valid": true) determinated by LLM, it proceeds by sending it to local planner to execute. If not valid, the LLM would tell path planning which area is not valid, and the path planning algorithm would plan a path without the invalid area.
8. move_base Planner: The move_base planner takes the results from the LLM and preceed. If unsuccessful due to door close or not enough space for the robot to go through, it report back, save this fail experience, go back to path planning algorithm to plan a new path.

And our main tasks can be divided into following parts:

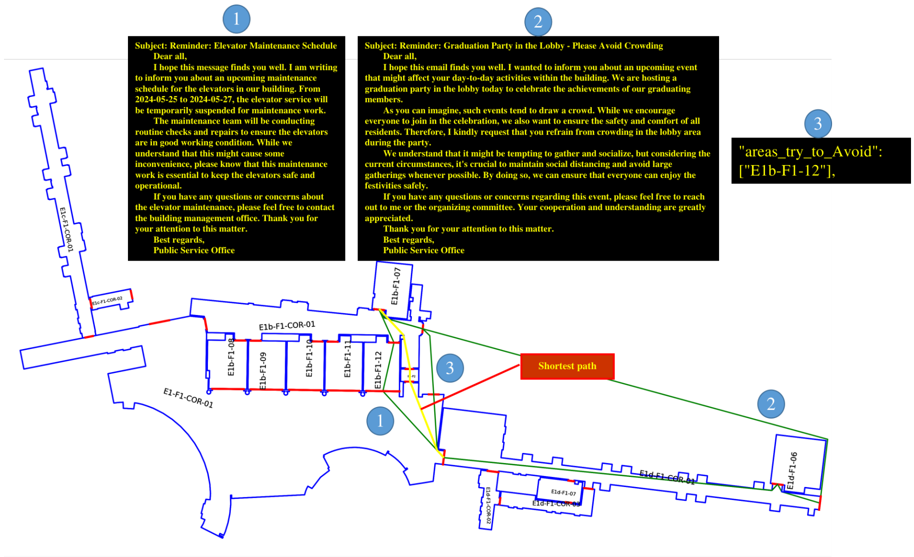

Figure 3. osmAG example

1. Align osmAG map with map used in ROS move_base.
2. Given path planned from LLM to global planner, integrate the path to "Base-GlobalPlanner" of move_base [1]
3. Utilize move_base's local planner to navigate through crowd or clutters.
4. Path may change during operation, the local planner needs to provide interface.
5. Provide response to LLM, for example, success or error message to record experience.

Our goal is to achieve intelligent path planning that could handle external textual information like news or e-mails, we have done some tests already, shown in Fig. 3. The yellow path in the image is the shortest path that a traditional A* algorithm gives. In the first example, a elevator maintenance is announced vis e-mail, LLM understands this information and tell the path planning algorithm to avoid this elevator. The second example is that a graduation party is schudeled in the lobby, therefore a lot of people would gather in the lobby, the LLM catches this information and avoid the lobby by taking a detour outside the building. The third example is that according to some general information, people know to try to avoid some room for example a VIP conference room, the LLM could capture this information and tell the path planning algorithm to try to avoid these rooms by adding path length.

4. System Evaluation

Our algorithm could plan a path based on the map and current situation, for example in the experient video, a notification of graduation party in SIST lobby would result in LLM to reject path including pass through the lobby, and the osmAG planner would use another path before execuate the navigation, where a vanilla move_base planner would first go through the lobby, find that is not accessible and turn back to find another path. In our project, the system is tested using gazebo, with a robot using vanilla move_base planner given a occupancy grid map and a robot

[1] http://wiki.ros.org/navigation/Tutorials/Writing

using osmAG and get informantion from external source and previous experience. The test is from room 'E1d-F1-13' to room 'E1b-F1-04', normally the path would include the lobby, but with external constrain like party in the lobby, the robot should navigate avoid the lobby.

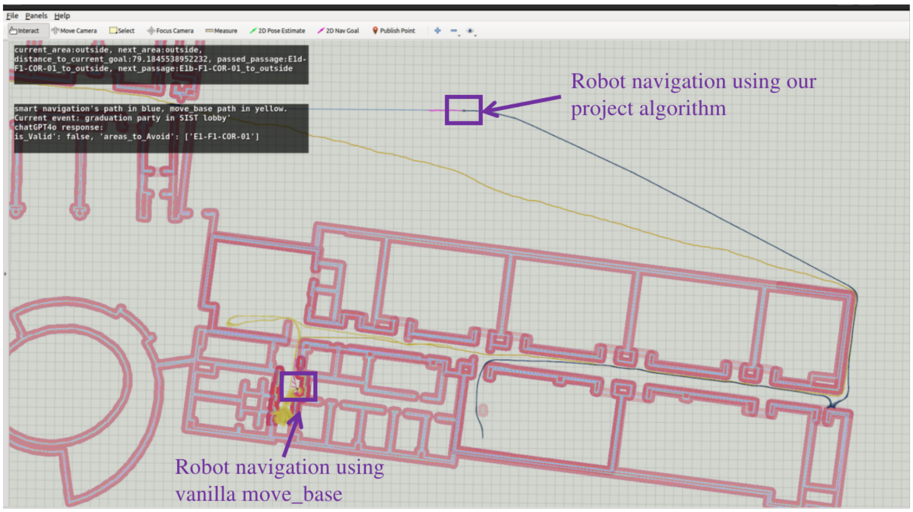

Figure 4. intelligent navigation experiment

5. How To

1. You need to have a opanai api key, put your key in config.json file
2. Properly build and source your workspace, and run

```
roslaunch mobile_manipulator_body base_gazebo_control_post_office_two_robot.launch
```

   this will start the gazebo world.
3. Then run

```
rosrun call_openai simple_command_move_base.py
```

   this script will send goals to both robot and start navigation.

6. Conclusions

In this project, we utilize osmAG, a semantic topometric and hierarchical map representation to achieve intelligent navigation not only receive distance information from Laser but also from external source with the help of LLM. Our approach not only exploit the intelligence from LLM but also maintain the robustness of classical robotic algorithms, which effectively ground the LLM to real world.

In the rapidly evolving field of artificial intelligence, LLMs are becoming increasingly prevalent, opening new opportunities for enhancing robots' intelligence. However, effectively leveraging LLMs in robotics to augment the intelligence of robots remains an active area of research. This project aims to introduce osmAG as a versatile map representation for future LLM-robot systems, designed to be interpretable by LLMs, compatible with traditional robotic algorithms, and understandable to humans.

In contrast to traditional robotics, which has been studied for decades, the exploration of integrating traditional robotic algorithms with Large Language Models (LLMs) is just beginning. With osmAG, a map representation that is understandable by LLMs, compatible with robotic algorithms, and comprehensible to humans, we aim to expedite this research process.

References

[1] D. Feng, C. Li, Y. Zhang, C. Yu, and S. Schwertfeger, “osmag: Hierarchical semantic topometric area graph maps in the osm format for mobile robotics,” in 2023 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2023, https://arxiv.org/pdf/2309.04791.pdf.

[2] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu et al., “Palm-e: An embodied multimodal language model,” arXiv preprint arXiv:2303.03378, 2023.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn et al., “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” arXiv preprint arXiv:2307.15818, 2023.

[4] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, “Task and motion planning with large language models for object rearrangement,” arXiv preprint arXiv:2303.06247, 2023.

[5] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, “Chatgpt for robotics: Design principles and model abilities,” Microsoft Auton. Syst. Robot. Res, vol. 2, p. 20, 2023.

[6] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, “Correcting robot plans with natural language feedback,” arXiv preprint arXiv:2204.05186, 2022.

[7] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, “Latte: Language trajectory transformer,” in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 7287–7294.

[8] X. Wang, W. Wang, J. Shao, and Y. Yang, “Lana: A language-capable navigator for instruction following and generation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19 048–19 058.

[9] Z. Li and H. Ning, “Autonomous gis: the next-generation ai-powered gis,” arXiv preprint arXiv:2305.06453, 2023.

[10] Z. Dai, A. Asgharivaskasi, T. Duong, S. Lin, M.-E. Tzes, G. Pappas, and N. Atanasov, “Optimal scene graph planning with large language model guidance,” arXiv preprint arXiv:2309.09182, 2023.