Language Embodied Navigation using Local and Global Planners

UdayGirish Maradana*
Department of Robotics Engineering
Worcester Polytechnic Institute
umaradana@wpi.edu

Venkata Sai Krishna*
Department of Robotics Engineering
Worcester Polytechnic Institute
vbodda@wpi.edu

Pradnya Sushil Shinde*
Department of Robotics Engineering
Worcester Polytechnic Institute
pshinde1@wpi.edu

Butchi Venkatesh Adari*
Department of Robotics Engineering
Worcester Polytechnic Institute
badari@wpi.edu

* Equal Contribution

***Abstract*—Language is something we humans use to connect with each other to express emotions/opinions, to delegate tasks. It's almost impossible for a human to give another a JSON file to execute something, so language is the primary aspect of everyday life. Our project inspiration started with how we can convert a task defined in Human language can be converted to a set of sequences of actions that can be understood by a Robot. The project is inspired by the FacebookAI Habitat Challenge - 2023/2022 Object Nav theme and also a recent presentation by Dr. Stefanie Tellex at the Robotics Colloquium (RBE). The objective is to develop agents that can navigate unfamiliar environments and move away from closed object classes towards open-vocabulary natural language. So the challenge is for a robot (virtual/real) when placed in an unknown environment should be able to navigate the environment by using information from its State space (From IMU + GPS/Encoders) and Image (visual cues) information.**

I. Introduction

Imagine waking up to a home robot and saying *"Hey Robot! Get me a coffee from the kitchen!"*. To achieve this, one will have to integrate the visual, navigation, and sensing capabilities to an extent such that the robot accurately interprets where and how to reach. Nowadays, Robots have the capability to achieve most of complex tasks such as Navigation and Task-based planning. But, for the Robots to be used with a human who does not code or knows the intricacies of the Robots, the barrier is language. Solving the problem of language in the case of communicating and getting the result by the continuous query has been achieved recently by solutions such as ChatGPT, and Transformer networks. Now the question is how we can use language as a tool to communicate with a Machine. This is what we want to address, as a part of this project where we use human language (English) as input to specify a machine to perform certain navigation or task. Primarily we want to address some approaches such as using language cues and executing a sequence of actions based on traditional navigation algorithms and compare that with some of the End-to-end approaches such as Behaviour Cloning and Reinforcement learning.

This is an active research area, where multiple challenges such as the Habitat Challenge and NeurIPS are concentrated upon.

II. Related Work

- PIRL Nav [5] studies Pretraining with Imitation and RL Finetuning for ObjectNav. This Object Nav is one of the theme in habitat challenge. This work is about exploring Imitiation learning using behaviour cloning (BC) on a dataset of human demonstrations achieves more promising results than pure Frontier exploration.
- Success Weighted by Completion Time [6]: A Dynamics-Aware Evaluation Criteria for Embodied Navigation. This paper concentrates on defining a new metric for navigation than the generally used classification based on time taken or path length which have flaws such as the least time taken might not be the correct path in case of a task with complex instructions to the robot nor the least path is correct in terms of time taken. To train agents or classify, a new metric was defined which is SCT (Success Weighted by Completion Time). SCT explicitly takes the agent's dynamics model into consideration, and aims to accurately capture how well the agent has approximated the fastest navigation behavior afforded by its dynamics. While several embodied navigation works use point-turn dynamics, this metric focus on unicycle-cart dynamics for our agent, which better exemplifies the dynamics model of popular mobile robotics platforms (e.g., LoCoBot, TurtleBot, Fetch, etc.). It enhances one of the previous metric SPL (Success weighted by Path Length) which is limited in its ability to properly evaluate agents with complex dynamics.
- Simple but Effective [1]: CLIP Embeddings for Embodied AI - Contrastive language image pretraining (CLIP) encoders have been shown to be beneficial for a range of visual tasks from classification and detection to captioning and image manipulation. This paper investigates the effectiveness of CLIP visual backbones for Embodied AI

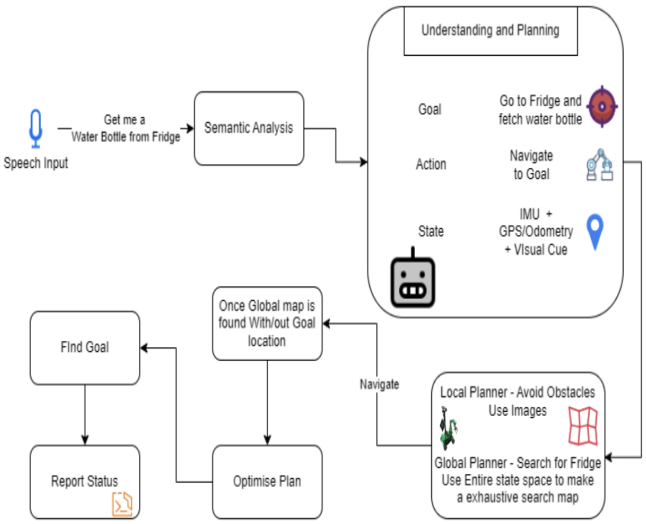


Fig. 1: Project Approach /Architecture

(a) Scene1

(b) Scene2

Fig. 2: HM3D Tested Environments

tasks. CLIP family of neural network are well known for their remarkable performance with zero shot learning which can match the accuracy of a fully trained ResNet50 on ImageNet.

III. Proposed Methods

We have established a platform that performs Object Navigation using the traditional algorithms of RRT^* and A^* as well as RL methods in the Habitat Simulation Environment. Object Navigation is implemented by commanding a speech input to the robot that understands the text and identifies

Fig. 3: Object Detection

the object in the text. The robot then navigates to the goal position where the object can be found. During this process, we receive multiple RGB and Depth image cues that can be perceived as the observations recorded by the robot while navigating to the goal position. In the scope of the project, we have selected the Habitat Matterplot 3D dataset for testing in various environments. The Habitat-Matterport 3D Research Dataset (HM3D) is the largest-ever dataset of 3D indoor spaces. It consists of 1,000 high-resolution 3D scans (or digital twins) of building-scale residential, commercial, and civic spaces generated from real-world environments. We propose the implementation of traditional planners in Model 00005-yPKGKBCyYx8 presented as Scene 1 and Model 00049-e3YKRHQRPNe presented as Scene 2 in Figure 2. Below is a detailed description of the individual components of the navigation process.

1) The initial stage is to convert a speech or language-based command to a text command that can be understood by high-level language through Semantic analysis. This is achieved by implementing a voice-to-intent procedure using a trained model. Following is a nested description:
   - **Speech Recognition**: Capture the audio input through the microphone.
   - **Text Preprocessing**: The recognized speech goes through text preprocessing that involves text cleaning and tailoring the text to a predefined length such that only the non-trivial information is taken into consideration.
   - **Intent Prediction**: The post-processing method deals with providing an intent output that can be presented as an object to navigate to.
2) Once we have an "Object Goal" to navigate to, we need to fetch the real-world coordinates of the object for the path planner to process. The next step therefore implements a procedure to retrieve goal coordinates.
   - **Simulator Configuration**: We provide a configuration for the simulator to process that describes various parameters such as scene details, camera sensor specifications, and agent's action space. The agent's action space consists of 'move forward', 'turn left', 'turn right' and 'stop'.

- **Top-Down Map Generation**: Using the simulator's pathfinding capabilities, we generate a top-down map image of the explored environment. The map subsequently describes the environment using a polygonal arrangement of each room in the scene.
- **Agent Initialization**: An agent is initialized with an initial state and position in the configured environment.
- **Random Spawning, Navigation, and Goal Detection**: The robot is spawned at different locations in the environment and allowed to explore. 20 to 30 random samples are generated in spawning. The robot uses a YOLO Object Detection model to identify the goal object. Once the visual sensing capabilities of the robot provide an object that can be identified as the goal, the position coordinates of the goal object are returned to the user.

3) After receiving the goal position coordinates of the object, we move on to planning in and around the environment. This requires a position for the robot to understand in a 3D environment and a position for the planner to sample in a 2D grid environment.

- **3D to 2D goal conversion**: The goal coordinates retrieved through random spawning of the robot describe a three-dimensional position. They are therefore converted to 2D representation by obtaining the Axis-ALigned Bounding Box of the navigation mesh. The 3D coordinates are normalized and scaled using the minimum and maximum corners of the AABB and a scaling parameter that determines how much each meter in 3D space corresponds to in the 2D top-down view.
- **Path Planning**: The 2D start and goal coordinates are passed to a traditional path planner along with a map array of the top-down map which is loaded for each active scene. The path planners then provide a set of tree nodes that lay out the optimal path to the goal position.
- **2D to 3D Conversion**: Since the traditional planners will provide us with 2D path points, we need to map these to the coordinates in the real world. An approach of conversion similar to 3D-to-2D is implemented. While converting, it is checked that floor elevation(height parameter in the path) is tuned such that the x and z coordinates majorly impact the path while the y coordinate(elevation in 3D space) shows minor variations. This is done to ensure a smoother trajectory traversal.
- **Visualization**: The traditional planner path points and sampled tree nodes are visualized along with the path laid out on the top-down map
- **Sensory Observations**: As the robot navigates to the in the environment, it collects visual sensory information that can be retrieved in the form of RGB and Depth images as seen in **Figure 4**.

RL Based Approaches	Traditional Approaches
1. End to End approach	1. Cascaded Approaches
2. Single Point of failure	2. Multi Point failure
3. Can result in a proper solution if have enough data	3. Does not depend much on data at least but unreliable in uncertain environments
4. More generic and learning based	4. Algorithmic and Depends on initialization etc.
5. Harder to achieve. Have more scope towards research.	5. Most of the algorithms already achieve a decent enough performance. Narrow scope to improve.

Fig. 4: RL Approaches vs Traditional Planners

IV. Platform & Evaluation

The evaluation criterion for the scope of this project is to understand the difference between a traditional method and RL /Imitation-based learning. The objective is to understand the applicability of traditional planners in real-world environments and to understand the feasibility of these planners.

We have also compared the performance of two state-of-the-art traditional planners: RRT* and A*. RRT* proves to be a master in finding an optimal given the number of nodes it samples and its capability to explore the environment using tree extension. On the other hand, while A* may not provide the shortest path, it does produce a smoother real-time trajectory compared to RRT*, which makes sense given the random sampling nature of RRT*.

We observed certain situations where RL performed better than Traditional Algorithms if there is adequate data to experiment, while traditional planners are much easier to implement and integrate. Figure 4 shows the differences between RL-based approaches and traditional planners.

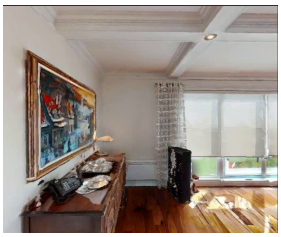
V. Limitations

When the view is shifted from the 2D grid world to the 3D complex world, there is a spike in problems that need to be solved. With the increase in dimensions, planning requires consideration of various factors that may limit the optimality of the planners. The following are limitations that we came across in the context of the problem statement:

1) Traditional planners such as RRT*, and Informed RRT* focus largely on achieving global optimality, which can be computationally expensive in real-world environments, therefore limiting their performance.
2) Lack of adaptability and ability to learn from failures reduce the efficiency of traditional path planners in uncertain environments.
3) Traditional methods produce paths that may suit a predefined environment however in a dynamic environment where the robot may have to recalculate the path based on the present situation, these planners might fail.

VI. Challenges

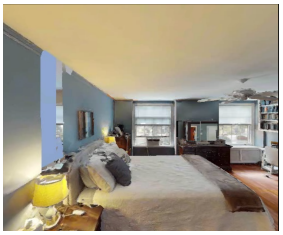
- One of the challenges we faced in developing the project was extracting three-dimensional path points for the robot



(a) Scene1, Start: Living Room

(b) Scene1, Goal: Sofa

(c) Scene2, Start: Living Room



(d) Scene2, Goal: Bed

(e) Scene 1 Depth, Goal: Sofa

(f) Scene2 Depth, Goal: Bed

Fig. 5: Visual sensor observations retrieved by Robot at start and goal positions

to interpret and perform. RRT* provided us with certain trajectories that went out of bounds due to a certain goal position in 3D navigable space.

- For producing a continuous trajectory, we were required to generate interpolated path points. This was an arduous task to complete given the calculations required in generating 3D paths.

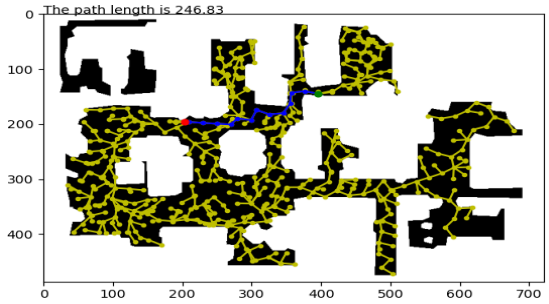
VII. Break-down contributions of team members

A. Exploration of Traditional Algorithms and Selection of Simulation Environment

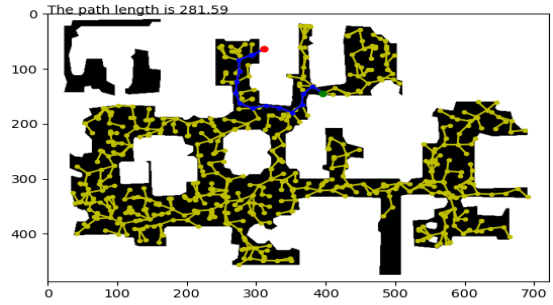
The objective of the task is to explore path-planning algorithms such as A^*, and RRT^* and evaluate their performance on their ability to behave as an asymptotically optimal planner. After careful evaluation, we have decided to implement RRT^* for the application of this project. This task is a contribution of **Uday**, **Krishna**, **Pradnya**, and **Venkatesh**.

B. Environment Setup

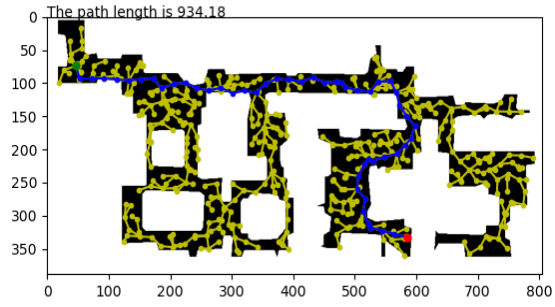
The objective of the task is to set the vision sensor feed, add a Navigation baseline, import a custom URDF file, and tweak the required parameters. Moreover, RL training and task exploration are also included as a part of this task.
Following is the contribution:



(a) Scene1, Refrigerator: RRT*



(b) Scene1, Sofa: RRT*



(c) Scene2, Bed: RRT*

Fig. 6: Traditional Planners

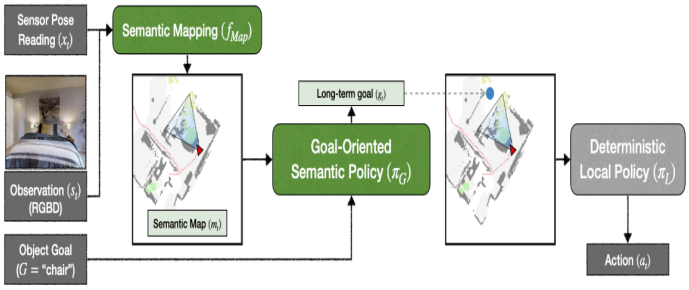
- Obtain and analyze vision sensor feed: Completed by Venkatesh
- Navigation baseline: Completed by Pradnya
- Custome URDF: Completed by Krishna
- RL training and task exploration: Completed by Uday

C. Natural Language Processing and Goal Retrieval

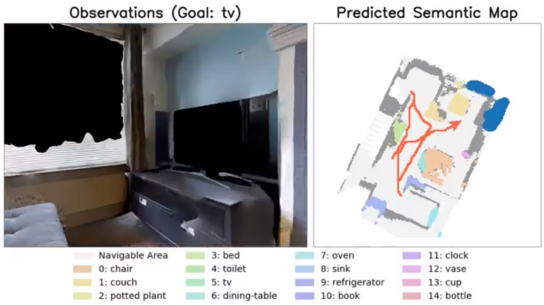
The objective of the task is to identify speech commands and process the text to understand the goal object in the statement. Further, once the goal object is known, a three-dimensional coordinates set is obtained for the traditional path planners to plan upon. This section has been developed by **Venkatesh**.

D. Path Planning

Once a set of coordinates of both a goal and start position in 3D is obtained, it needs to be converted to a 2D set of coordinates. These 2D coordinates are then passed to traditional planners, RRT* and A* which search for the goal



(a) RL Workflow



(b) RL based planning: "Go to TV"

Fig. 7: RL Implementation

coordinates in the grid environment of the active scene. The 2D path points obtained by the planners are converted to 3D. While navigating the 3D path points, the robot retrieves RGB and depth information. The results obtained for RRT* can be seen in **Figure 6** for two scenes and three goal objects, 'Sofa', 'Refrigerator', and 'Bed'. The top-down map of the environment has been plotted on a grid with black space defined as free space and white space as obstacle space. This section has been developed by **Pradnya**.

E. RL based Learning

Since the objective of the project is to compare traditional planners with Reinforcement learning-based planning we trained an RL Model to navigate in unknown environments and reach specified goal object positions. The implementation workflow and results can be seen in **Figure 7**. This section has been developed by **Uday**.

F. Project Integration

To obtain goal-specific results, it was important to arrange each task in individual classes and integrate it into a single executable file. Function integration and results retrieval were done by **Krishna**.

ACKNOWLEDGMENT

We are thankful to Dr. Constantinos Chamzas for giving us the feasibility of forming a big team and working on a exploratory real world problem as a part of the Motion planning coursework.

REFERENCES

[1] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but Effective: CLIP Embeddings for Embodied AI," 2022.
[2] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," 2019.
[3] J. Gu, D. S. Chaplot, H. Su, and J. Malik, "Multi-skill Mobile Manipulation for Object Rearrangement," 2022.
[4] S. Yenamandra et al., "The HomeRobot Open Vocab Mobile Manipulation Challenge," in Thirty-seventh Conference on Neural Information Processing Systems: Competition Track, 2023.
[5] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "PIRLNav: Pretraining with Imitation and RL Finetuning for ObjectNav," in CVPR, 2023.
[6] N. Yokoyama, S. Ha, and D. Batra, 'Success Weighted by Completion Time: A Dynamics-Aware Evaluation Criteria for Embodied Navigation', arXiv [cs.RO]. 2023.
[7] Chaplot, D.S., Gandhi, D., Gupta, A. and Salakhutdinov, R., 2020. Object Goal Navigation using Goal-Oriented Semantic Exploration. In Neural Information Processing Systems (NeurIPS-20).