Objective

Teams are expected to come up with a vision-language model that can take a natural language navigation query and navigate the vehicle system by generating a waypoint or path based on the query. The navigation will be done in a custom simulation environment for 2024.

System Setup

Follow the docker instructions to try the simulator setup. An Ubuntu computer is highly recommended for the setup as the docker image is prepared based on Ubuntu 20.04 and ROS Noetic. Pull the docker image provided and launch the system in it.

The system has two parts both in the home folder of the docker images:

- The base navigation system is in the '*cmu_vla_challenge_unity'* folder. For the base navigation system, you may change the scene in the '*src/vehicle_simulator/mesh/unity*' directory. A set of 15 environment models can be downloaded from here.
- The vision-language model is in the '*AI_module*' folder. The model currently in the folder is a dummy model and teams are expected to come up with an alternative model to replace this one.

Launching the system startup script '*start_cmu_vla_challenge.sh*' in the home folder, the dummy model will send visualization markers for object reference and waypoints to guide vehicle navigation. The two types of messages are listed below. To integrate the users' model with the system, please modify the system startup script.

- Visualization marker: ROS Marker message on topic name: /selected_object_marker, containing object label and bounding box of the selected object.
- Waypoint: ROS Pose2D message on topic /way_point_with_heading (neglect the heading for this year's challenge).

Task Specification

PDF files with 5 questions for each scene and the expected responses are provided to the teams. The questions are from 3 categories - numerical, object reference, and instructions following. For numerical questions, we expect the team's software to print a number in the terminal. For object reference questions, we expect the teams' software to publish a visualization marker (similar to the dummy model) to highlight the referred object. For instruction following questions, we expect the team's software to send waypoints to guide the vehicle navigation (also similar to the dummy model). We provide a reference trajectory file along with the PDF files for each instruction following question. To ensure all teams are competing in the same base navigation environment, the current settings in the '*cmu_vla_challenge_unity*' folder should be kept the same and only the environment model inside it switched.

The method developed is required to take in natural language questions similar to the ones listed in the PDF file. To develop software for completing the task, any information provided by the simulation system is allowed to be used. During evaluation, however, the method developed is only allowed to use the following ROS messages, as if integrating with a real-world robot system:

Message	Description	Frequency	Frame	ROS Topic Name
Image	ROS Image message from the 360 camera. The image is at 1920/640 resolution with 360 deg HFOV and 120VFOV.	10Hz	camera	/camera/image
Registered Scan	ROS PointCloud2 message from the 3D lidar and registered by the state estimation module.	5Hz	map	/registered_scan
Sensor Scan	ROS PointCloud2 message from the 3D lidar.	5Hz	sensor_at_scan	/sensor_scan
Local Terrain Map	ROS PointCloud2 message from the terrain analysis module around the vehicle.	5Hz	map	/terrain_map (5m around the vehicle) /terrain_map_ext (20m around the vehicle)
Sensor Pose	ROS Odometry message from the state estimation module.	100-200Hz	from map to sensor	/state_estimation
Traversable Area	ROS PointCloud2 message containing the traversable area of the entire environment.	5Hz	map	/traversable_area
Ground-Truth Semantics	ROS MarkerArray message containing object labels and bounding boxes within 2m around the vehicle.	5Hz	map	/object_markers

Submission

Follow the docker instructions to commit, tag, and push your docker image to DockerHub and make it public. The source code may be removed and only the executable left. Make sure to download the docker image and test it following the instructions because the same way will be used for the challenge evaluation. Further, please make sure to **send the waypoints and visualization markers of the same types and on the same ROS topics as in the dummy model** so that the base navigation system can receive them. For submission, send us the link ([docker_id/repository_name:tag_name]) via email to haochen4@andrew.cmu.edu by the deadline specified on the website or fill out the submission form (posted soon). **In the email, explicitly note if your AI module uses the ground-truth semantics.**

Evaluation

The submitted docker image will be pulled and 3 environment models which have not been publicly released will be used to evaluate it. For each scene, 5 questions similar to those in the provided PDF files will be tested and a score given to each of the responses. For numerical questions, we expect a number printed in the terminal. A score of 0 or 1 will be given. For object reference questions, we expect a ROS visualization marker shown in RVIZ. A score of 0 or 1 will be given depending on if the visualization marker's center point is within 2m relative to the ground truth object's center point in X-Y. For instruction following questions, we expect the vehicle navigation guided by waypoints. A score will be given as the points gained from the navigation (0, 1, 2, or 3) minus a penalty, where the penalty is summated over the n trajectory points as described in the equations below. The term dis_i^{min} is the minimum distance from the trajectory point to the given reference trajectory (along with the PDF files). As a result, the longer the duration and the larger the deviation to the reference trajectory, the higher the penalty. For object approaching in the instruction following, we use 2m as the relative distance requirement (in X-Y), the same as the object reference questions. Each team's overall score will be the sum of the scores for all the questions in the 3 scenes.

$$score = points - penalty,$$

$$\text{where } penalty = 0.01 \sum_{i=0}^{n} dis_i^{min} \Delta t.$$

Registration

Please fill out the registration form on the challenge website to register your team. Any questions or concerns can be sent by email to haochen4@andrew.cmu.edu.

Additional Information

Challenge website: https://www.ai-meets-autonomy.com/cmu-vla-challenge

Base navigation system repository with Unity scenes:
https://github.com/jizhang-cmu/cmu_vla_challenge_unity

Base navigation system repository with Matterport scenes:
https://github.com/jizhang-cmu/cmu_vla_challenge_matterport