2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Language Enabled Hierarchical Scene Graphs For Precision Agriculture Autonomy

Adam Mukuddem, Adam Speed-Andrews and Paul Amayo

***Abstract*— The focus on human-robot collaboration has emerged as a pivotal area in the advancement of precision agricultural systems [1]. This strategy exploits the distinct strengths of both humans and robots while minimising the exertion of each [2]. A central aim within human-robot collaboration is to create robotic systems that are capable of understanding instructions given in natural language. Agricultural settings, especially those with structured rows of crops, are characteristically uniform, presenting difficulties in accurately grounding instructions and navigating the space. In this paper, we establish a systematic method for robotic platforms operating within agricultural settings to recognize natural language directives and autonomously traverse toward specified targets, gathering data en route. We advance the 3D Scene graph model introduced in *Osiris* [3], adapting it to support autonomy through a Visual Teach and Repeat paradigm, which does not rely on an expansive navigation stack. Additionally, we exploit large language models to correctly ground instructions within the newly constructed 3D scene graph representation, thus enabling natural language directives to be relayed to robotic systems in agricultural contexts. The system's ability to interpret and execute natural language commands is confirmed through validation and evaluation in a practical agricultural scenario via a ground robot.**

I. Introduction

Imagine the scenario where humans carry out complex tasks and robots assist with labour-intensive, repetitive, and arduous tasks in precision agricultural environments. This approach maximises the unique capabilities of both humans and robots and minimises the effort of both parties [2], however, performance in these precision agricultural tasks is limited by the extent of collaboration between the two. As such, improving human-robot collaboration has been identified as one of the key focus areas for the development of precision agricultural systems [1]. Concurrently, there has been a growing effort to create versatile robotic systems designed to process instructions given as natural language. These include systems like SayPlan [4], DELTA [5], and CLIO [6], which are adept at understanding natural language commands. These systems are designed for use by robots *exploring* indoor and urban outdoor settings where objects and terrain can be clearly distinguished. Agricultural environments, in particular, rowed crops, are inherently homogeneous [7], which presents challenges in the ability to ground instructions accurately. Their associated tasks are also innately repetitive, placing a higher emphasis on repeated actions rather than unbounded exploration. These environments also experience rapid changes throughout the day due to weather changes, lighting changes, and agricultural activities such as harvesting, further increasing the localisation burden as tasks are being repeated.

Adam Mukuddem, Adam Speed-Andrews and Paul Amayo are with the African Robotics Unit, University of Cape Town, South Africa.

Fig. 1: The figure illustrates the Husky A200 autonomously navigating an agricultural environment following a natural language command.

In this work, we propose a systematic approach which enables robotic platforms monitoring and working in agricultural environments to receive instructions in natural language and autonomously navigate to requested locations collecting data across its path. We take advantage of the 3D Scene graph structure developed in [3] of an agricultural environment and develop a novel addressing system that facilitates the grounding of instructions within a homogeneous agricultural environment. We then leverage the capabilities of language models for grounding instructions within the generated 3D scene graph representation, and thus enable natural language instructions to be issued to a robotic platform within agricultural environments. We incorporate a Visual Teach and Repeat paradigm to enable autonomy without the need for an expansive navigation stack and overcome the challenge of free space not translating to traversability within agricultural environments.

We summarise the main contributions in this paper as follows:

- We develop an addressing system for homogeneous agricultural environments to enable grounding of instructions
- We develop a novel system which integrates Large Language Models and 3D scene graphs to facilitate issuing natural language instructions to robots within homogeneous agricultural environments
- We leverage a teach and repeat framework for autonomy to overcome the challenge of free space not relating to traversability and bypass the need to generate instructions in a planning language
- We validate and evaluate the performance of the system using real-world data collected in agricultural environments and on-field experiments.

The paper is organised as follows. In Section II related

work is discussed. In Section III, we present an overview of the construction of the 3D scene graph. In Section IV we explain how the system grounds natural language instructions. In Section V we explain how we produce a navigable plan from the output of the LLM. Section VI showcases the performance of the system on real-world farm data collected. Finally, conclusions are drawn in Section VII.

II. Related Work

A. 3D Scene Graphs

Three-dimensional scene graphs offer a robust and intuitive means to represent intricate 3D settings [8]. These graphs convey environments through a hierarchical or layered structure, where the nodes signify various spatial concepts, from basic geometry to comprehensive interpretations at the scene level, and the edges illustrate relationships [8]. The organisation of the graph layers is tailored to the architecture of the environment, taking into account task and motion planning queries [8]. Predominantly, research on 3D scene graphs has focused on indoor environments [9], [10]. Strader *et al.* [11] introduced a method for developing 3D scene graphs that can be applied to indoor and outdoor environments. They noted that while indoor environments have a clear hierarchy of semantic concepts, such clarity is less apparent in outdoor settings. Osiris [3] extended the concept of a 3D scene graph to agricultural outdoor settings. The 3D scene graph of agricultural environments consisted of layers Rows, Planting Lines, Plants and 3D Metric semantic mesh.

B. Robot Task Planning with Large Language Models

There has been a drive to develop robotic systems that integrate Large Language Models (LLMs). SayPlan [4] is a body of work that allows humans to give commands to a robot through a natural language instruction. SayPlan combines the power of a 3D scene graph together with a LLM. SayPlan exploits the hierarchical nature of the 3D scene graph. Given a list of tasks, the system generates a subgraph of nodes at each level of the hierarchial representation that is relevant to the list of tasks. Works implementing task planning using LLMs predominantly use 3D scene graphs of the environment [4]–[6]. Liu *et al.* propose DELTA [5], which uses LLMs for task planning coupled with 3D scene graphs as an environment representation. DELTA focuses on long-term planning by decomposing long-term task goals, which contain multiple instructions, into sub-goals. The 3D scene graphs are in the format of nested dictionaries in Python [5]. LLM-GenPlan [12] uses a LLM to summarise domain knowledge from input planning domain definition language (PDDL) files and then proposes a strategy to solve the problem.

In works that integrate 3D scene graphs with Large Language Models (LLMs) [4]–[6], [13], it has been demonstrated that natural language instructions can be effectively interpreted and executed by robots operating in indoor environments. However, applying these methods directly to outdoor agricultural environments presents unique challenges. Unlike indoor environments, which contain diverse and semantically distinguishable spaces (e.g., kitchens, bedrooms, hallways, as shown in Figure 2c and 2d), agricultural environments are often highly homogeneous. As illustrated in Figure 2a and 2b, a typical section of a farm consists of multiple planting rows with identical crops and minimal visual or semantic differentiation between them [7]. In indoor settings, task grounding can rely on contextual inference, such as deducing that an object associated with cooking is likely to be found in a room labelled 'kitchen,' as demonstrated in [4]. Similarly, Shah *et al.* [14] leverages known landmarks within structured environments to narrow the search space for tasks-relevant objects and locations.

These forms of inference, however, are not directly applicable in homogeneous agricultural settings, where rows of crops lack unique identifiers or contextual cues. This fundamental structural difference limits the effectiveness of methods developed for indoor environments. Therefore, while the combination of scene graphs and LLMs shows promise for intuitive robot instruction, new approaches are required to adapt these techniques for the unique spatial and semantic characteristics of outdoor agricultural environments.

The exploration of using LLMs as embodied agents for robot task planning in outdoor environments is limited. However, works such as Xie *et al.* [15], which explores the notion of altering LLMs for outdoor scenarios and Zuzuarregui *et al.* [16], which explores the use of LLMs within agricultural environments, demonstrate a growing research trend to the adaptation of LLMs for outdoor environments. Zuzuarregui *et al.* [16] developed an end-to-end system which allows for the issuing of natural language instructions to robots within an agricultural environment; however the limitations that the system fails with tasks requiring spatial awareness present an open research problem which can be explored within this work. Zuzuarregui *et al.* [16] use the Nav2 Ros2 autonomous navigation framework to navigate within an environment. In the works by Zuzuarregui *et al.* [16] and Strader *et al.* [17], they convert natural language instructions into Planning Domain Definition Language using an LLM. Comparatively, in our work, by ensuring that the natural language input is condensed into the created scene graph, the act of parsing the scene graph by an LLM creates directly executable instructions for autonomy via a teach and repeat framework.

C. Autonomous Navigation

Ravichandran et al. [18] developed a reinforcement learning framework for learning navigation policies on 3D scene graphs developed in [9]. The graph neural network generated derives high-level navigation instructions based on a 3D scene graph input. In [18] an action layer is added to the graph structure presented in [9]. The action layer represents the immediate space around the robot and creates a bridge between the scene graph and the low-level action space of the robot [18]. Nodes of the action layer embed semantic and traversability information and edges between action layer nodes and the scene graph are only added if a series of rigorous checks are passed, which determine if there is free space for traversability [18]. The navigation policy achieved reasonable results, but the policy is complex and requires several checks hard coded relative to the indoor structure wherein experiments were carried out. It is important to note

(a) Row 1 **(b)** Row 2 **(c)** Kitchen **(d)** Bedroom

Fig. 2: The figure illustrates the contrast between high-level semantic concepts found in outdoor agricultural environments and those in indoor settings. In outdoor agricultural environments, 'rows' often include numerous similar plant objects, whereas indoor spaces like 'kitchens' and 'bedrooms' present a diverse and sometimes distinct range of lower-level concepts, useful for their differentiation.

that traversability is *estimated* in this framework.

Gaspirino *et al.* [19] developed a navigation system that allows the switch of different sensing modalities (GNSS and camera-based navigation) used for localisation and navigation purposes within agricultural environments. Camera-based navigation is used to navigate within crop rows and GNSS is used to navigate between rows. Kim *et al.* [20] developed a system that incorporates a 3D LiDar in the navigation pipeline. The system consists of four modes for robust long-term navigation, which are row entering, in-row navigation, in- and out row classifier and row switching. These purpose-built agricultural navigation systems have demonstrated promising results; however, they require complex autonomy methods with varying switching modes to cater for agricultural environments.

While Visual Teach & Repeat (VT&R) approaches have not been adapted to scene graphs, they have remained an effective and simpler solution for autonomous navigation [21]–[24]. In these approaches, a robot is driven along a route of interest, with information being recorded along the path, to create what is known as the teach pass. During the teach pass, visual features are triangulated to form 3D landmarks. In the repeat pass, incoming image features are matched to these landmarks, and in this way, the estimate of the robot's position in relation to the teach pass can be obtained. This estimate can then be used to generate instructions to correct the robot's trajectory to match the teaching pass. Within this work we show how using a VT&R approach within a scene graph can lead to verifiable autonomy.

III. 3D Scene Graphs Construction

This section describes the adjusted concept of a 3D scene graph for agricultural environments, based on [3]. Our framework more explicitly enables the ability to ground precision agricultural instructions through the introduction of a novel addressing system. We also introduce a *robot* layer to more explicitly enable autonomy.

Formally defined, the hierarchical graph obtained through our proposed framework is $\mathcal{O} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges between nodes $\mathcal{V}$. The set of nodes $\mathcal{V}$ can be partitioned into the six layers of the graph as follows. $\mathcal{V} = \cup_{i=1}^{6} \mathcal{V}^i$.

The layers of this are farm sections, rows, planting lines, plants, robots and a 3D metric semantic mesh. *Layer 1* is a metric semantic 3D mesh. *Layer 2* is a topo-metric graph of the robot and selected sensor data for robot localisation, loop-closing, mesh merging and for calculating the control instructions for the teach-and-repeat autonomous navigation. *Layer 3* is a subgraph of the plant objects detected within the farm. *Layer 4* is a subgraph of the planting lines. Planting lines represent the lines in which the crops are planted. *Layer 5* is a subgraph of the rows within the farm. A row is defined as a robot-navigable area located in between two planting lines, in this way two planting lines form the lower layer of a specific row. *Layer 6* is a subgraph of the sections within the farm. Sections are areas of the farm for which the crops are of a specific type, as a farm can contain multiple crops.

In alignment with [3], our approach utilizes RGB images and depth data to construct a 3D scene graph. Initially, the system processes these inputs through a 2D perception stage, implementing visual odometry and semantic segmentation, as illustrated in Figure 3. The outputs are then used by a 3D perception system that constructs the metric-semantic mesh, identified as *Layer 1*, and further segments it to identify plant objects, forming *Layer 3*. The detected plant objects allow for the generation of planting lines via the Coral algorithm [25], which extracts geometric lines from the plant object data, resulting in *Layer 4*. Subsequent grouping of these lines produces robot-navigable rows in *Layer 5*, which are organized into sections containing uniform produce in *Layer 6*.

A key difference in our proposed approach is in how the *Layer 2* is constructed. In Osiris [3], this was constructed using the free space between the mesh, this free-space denoted by a path graph showed the navigable area that a robot could *potentially* move in. In this proposed approach, we adapt this to include the *actual* path that a robot has navigated through previously, this privileged path gives us a safe pathway to autonomy when used in a Visual Teach And Repeat paradigm. The following section provides more detail about the creation of this layer.

A. Robot Layer

The robot layer or *Layer 2* consists of a spatio-temporal graph created as the robot traverses the agricultural environment and we define it as an undirected graph consisting of pose and spatial vertices $\mathcal{V}^2 = \{\mathbf{X}, \mathbf{S}\}$ and their temporal and spatial edges. This is illustrated in Figure 5.

Pose vertices correspond to the initial keyframe position, its associated image and Bag of Words descriptors. The addition of the image and descriptors is critical for enabling of the localisation, subsequent aggregation and autonomy. Formally each pose vertex is defined as $\mathbf{X}_i^n = \{\mathbf{R}_i^X, \mathbf{t}_i^X, \mathbf{I}_i^X, \mathbf{D}_i^X\}$ where i is an index incremented as a new keyframe is

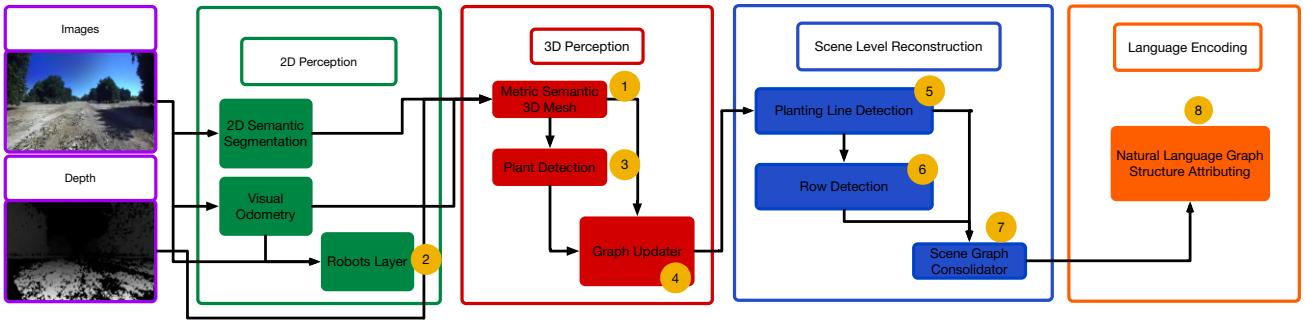


Fig. 3: Figure illustrating the process flow of the proposed framework. From a stream of RGB images and a depth estimate obtained from the ZED camera, our approach is able to create a hierarchical scene graph by performing 2D then 3D perception followed by reasoning over the entire scene. Natural language attributes are then encoded into each node. Stereo Images from Zed are also captured for the VT&R paradigm.

observed while n is a unique identification number given to each robot to facilitate further localisation tasks.

Temporal edges are added between the vertices and correspond to SE(3) transformations obtained through Odometry. These pose vertices and their corresponding edges are similar to classical pose graphs as seen within the SLAM literature, we enrich this formulation to allow for joint optimisation of the pose-graph and the underlying mesh using techniques commonly found in computer graphics [26] and adapted for robotics use-cases in Kimera-Multi [27].

We obtain this extra spatial information by sub-sampling the metric-semantic mesh *Layer 1* as it is observed in each new keyframe. This sub-sampled mesh is then simplified using a vertex clustering method that stores these vertices in an octree and then merges vertices belonging to the same voxel as the map grows. These merged vertices thus become the spatial vertices of the robot layer and are immediately connected to the pose vertex of its associated keyframe through a spatial edge. Each of these spatial vertices is defined as $\mathbf{S}_k = (\mathbf{R}_k^S, \mathbf{t}_k^S)$ where $\mathbf{R}_k^S \in SO(3)$ which is initialised to identity and $\mathbf{t}_k^S \in \mathbb{R}3$ is the 3D position of the merged mesh vertices. With this layer constructed as described, a complete open-loop hierarchical graph can be created for each individual robot, the following sections describe our proposed approach to aggregate these scene graphs. Additionally when objects are detected to form *Layer 3* edges are added between the object centroids and the pose vertices of the robot layer.

B. Scene Graph Output Format

The distinct semantic nature of objects within indoor environments allow for inference of the type of room within an indoor environment, based on the objects within a room. Within rowed crops there is inherent homogeneity between each line of crops and thus, there are minimal semantic features to differentiate one row from another. To address this issue we apply an enumeration system to all levels of the agricultural scene graph.

Given a set of plants in a planting line $\mathcal{P} = P_1, P_2, ..., P_n$, we establish an origin $\mathcal{O}$ for this planting line by taking the average of the x positions and using $y = 0$. We then calculate the distance $\mathcal{D}$ for each plant in the planting line from the $\mathcal{O}$. The plants within the planting are then assigned an attribute according to their distance from $\mathcal{O}$. The order is then linked

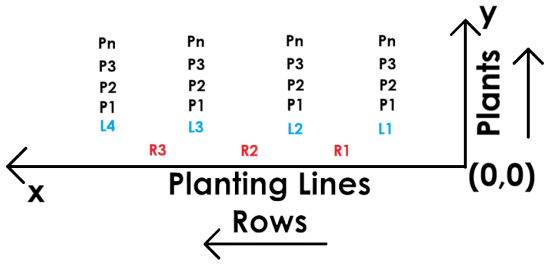


Fig. 4: Coordinate System of Plants, Planting Lines and Rows.

together with the Keyframe number, from the *Robot Layer*, to produce a *Natural Language* attribute which contains the Plant number and keyframe number. Linking the keyframe number to the plant number allows multiple plants with the same order number but within different planting lines to have a unique *Natural Language* attribute.

$$P_{NLA} = Plant\ N\ Keyframe\ X \tag{1}$$

Given a set of Planting Lines $\mathcal{L} = L_1, L_2, ..., L_n$, we order the planting lines according to the x coordinate. The x coordinate of the planting is the average of the x coordinates of the plant nodes within each planting line.

$$L_{NLA} = Line\ N \tag{2}$$

Similarly, given a set of Rows, $\mathcal{R} = R_1, R_2, ..., R_n$, we order the rows according the x coordinate. The x coordinate of the row is the average of the x coordinates of the planting lines within each row.

$$R_{NLA} = Row\ N \tag{3}$$

IV. Language Based Navigation

This section describes the approach we use to take a natural language input $\mathcal{I}$, the grounded scene graph $\mathcal{HO}$ to produce a robot navigable plan $\mathcal{A}$ for the monitoring of agricultural environments. Formally this can be defined as $\mathcal{A} = LLM(\mathcal{I}, \mathcal{HO})$ and the plan is then passed on for lower-level execution. While there are similarities to the general purpose approach presented in SayPlan, a key distinction is that agricultural environments tend to be more homogenous as compared to the indoor scenes where SayPlan operates.

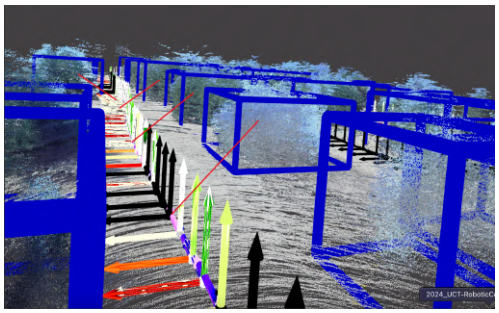


Fig. 5: The robot layer is comprised of keyframe poses which show the position and pose of a robot at each keyframe. Edges are drawn between the robot layer nodes and objects which were detected in a particular keyframe.

This homogeneity means that semantic objects can not be easily differentiated from each other, for instance one row of plants is semantically similar to the next row of plants while two adjacent rooms in a house even when serving the same function might differ in organisation, and/or objects present. To tackle this, semantic concepts are enumerated within the scene graph and we leverage these number allocations when issuing instructions.

Similarly to SayPlan we break down the processing of the instructions into two parts, firstly a Semantic Search within the graph followed by an iterative replanning.

A. *Semantic Search*

Even as scene graphs of agricultural environments are more homogenous when compared to their indoor counterparts, the scene graph itself can grow infinitely as the farm environment grows and the plants, lines, rows correspondingly increase. The first step of the instruction process is to identify the task-relevant portion of the scene-graph over which the language model should reason across.

This task-relevant portion of the graph $\mathcal{HO}'$ can be obtained by exploiting the hierarchical structure of the graph guided by the particular instructions. We first use the language model to perform spatial reasoning given by the instruction, an example input could be to "inspect row 4 and the 3 rows following it". This reveals a list of sub-tasks $\mathcal{S} = S_0, \cdots, S_n$ to be completed. For each sub-task a pruned scene graph is obtained based on the highest level node present in the instruction. From each pruned scene graph the required node to navigate to can be thus found by the LLM. With this node identified the planning stage can now begin. An example of a prompt for this is shown in the following text.

Agent Role:
You are a robot with spatial reasoning within a farm and you are tasked to inspect plants, lines and rows within a farm. You must break down the task into relevant sub-tasks before identifying the relevant key-frames which are linked to the plants,lines and rows in question.

Instruction: Move to Plant 0 in line 1 and then Plant 4 in line 4

Scene Graph: nodes: "Section 0":"Row 0, Row 1, Row 2", "Row 0":"Line 1, Line 2","Row 1":"Line 2, Line 3", "Row 2":"Line 3, Line 4", **"Line 1"**:"Plant 0 Keyframe 3, Plant 1 Keyframe 4, Plant 2 Keyframe 17, Plant 6 Keyframe 19, Plant 2 Keyframe 30, Plant 5 Keyframe 57, Plant 9 Keyframe 79, Plant 6 Keyframe 83, Plant 7 Keyframe 94, Plant 2 Keyframe 17, Plant 6 Keyframe 19, Plant 15 Keyframe 111, Plant 16 Keyframe 112, Plant 14 Keyframe 113, Plant 13 Keyframe 127, Plant 12 Keyframe 154, Plant 11 Keyframe 168","**Line 2**":"Plant 1 Keyframe 1, Plant 0 Keyframe 13, Plant 2 Keyframe 25, Plant 3 Keyframe 43, Plant 4 Keyframe 92, Plant 5 Keyframe 57, Plant 9 Keyframe 79, Plant 6 Keyframe 83, Plant 7 Keyframe 94, Plant 10 Keyframe 192, Plant 11 Keyframe 194, Plant 8 Keyframe 199","Line 3":"Plant 0 Keyframe 3, Plant 1 Keyframe 4, Plant 2 Keyframe 30, Plant 5 Keyframe 118, Plant 3 Keyframe 129, Plant 4 Keyframe 137","Line 4":"Plant 1 Keyframe 1, Plant 0 Keyframe 13, Plant 2 Keyframe 25, Plant 3 Keyframe 43, Plant 4 Keyframe 92"

Answer:
Keyframe 3,92

B. *Iterative planning*

Having received the list of nodes that the robot needs to navigate to, the objective of the iterative planner is to create a robot navigable plan $\mathcal{A}$ from this. To do so the optimised robot layer of the scene-graph is accessed. This reveals a topo-metric graph consisting of pose vertices and the edges between them, with these pose vertices intrinsically linked to the semantic nodes being navigated to, the iterative planner converts the $goto(node)$ command to a set of pose vertices $\mathbf{X}$ that the robot must navigate to. For object nodes this corresponds to the first pose from which the object was observed, for higher level nodes such as lines, rows and sections, this reveals the first and last pose vertices connected to these nodes. This set of vertices form the target vertices $\mathbf{X^T}$.

The robot must then also be localised for the planning action to complete, again this relies on the optimised robot layer of the scene graph and returns the current pose vertex $\mathbf{X^c}$ that the robot is occupying currently. Then a graph search is performed on the topometric robot layer $\mathcal{V}^2$ to reveal the sequences of vertices from the current pose to the target pose(s). This sequence of poses forms the trajectory the robot must navigate through. For navigational safety purposes the search can only proceed in the direction of forward robot motion, this ensures that any plans passed on have been previously navigated through the aid of an expert.

If no forward path exists to the target nodes this is reported to the user, if one exists this is passed on to the robot low-level controller. This robot navigable plan $\mathcal{A}$ can be thought of as a sub-map which after being taught by an expert, the autonomous controller must repeat. Section V details the controller used for this aspect.

As we leverage a teach and repeat framework for autonomy, the output of the LLM is only a set of keyframe(s) which limits the output token size.

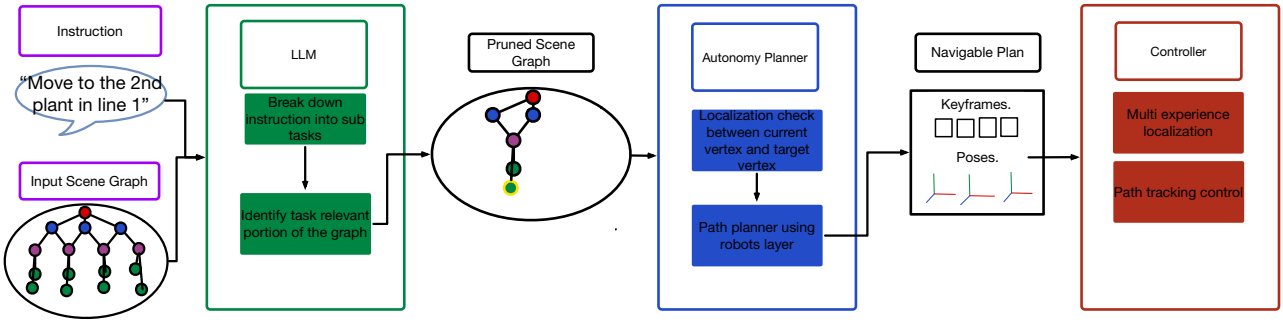


Fig. 6: Figure illustrating the process flow of the Language based Navigation pipeline IV. IV takes in a prompt and 3D scene graph which is passed into a Large Language Model (LLM). The LLM interprets the prompt and the scene graph and outputs a pruned scene graph with only task relevant nodes. The repeat planner then generates the navigable to the end goal as per the prompt. The controller then actions the navigable plan through multi experience localisation and path tracking control.

(a) Sunny Midday **(b)** Sunny Afternoon **(c)** Overcast

Fig. 7: Pictures onboard the Husky under different weather and lighting conditions

V. Autonomy

We use the VT&R3 navigation framework to build a network of traversable paths [21]. Incorporating the multi-experience extension helps to mitigate appearance changes in an agricultural environment caused by farming activities and normal produce growth over time. We use the information (Stereo Images and Odometry) collected during scene graph building to build a Spatio-Temporal Pose Graph (STPG) [24] using triangulated ORB features. We then use the Multi-experience localisation algorithm from [24] to support the autonomy. This algorithm estimates the posterior transform between the currently observed vertex in a repeat run to the closest estimated vertex from the teach path. A control action is computed from a path tracking controller to align the repeat run with the teach run.

VI. Experimental Evaluation

A. Experiments

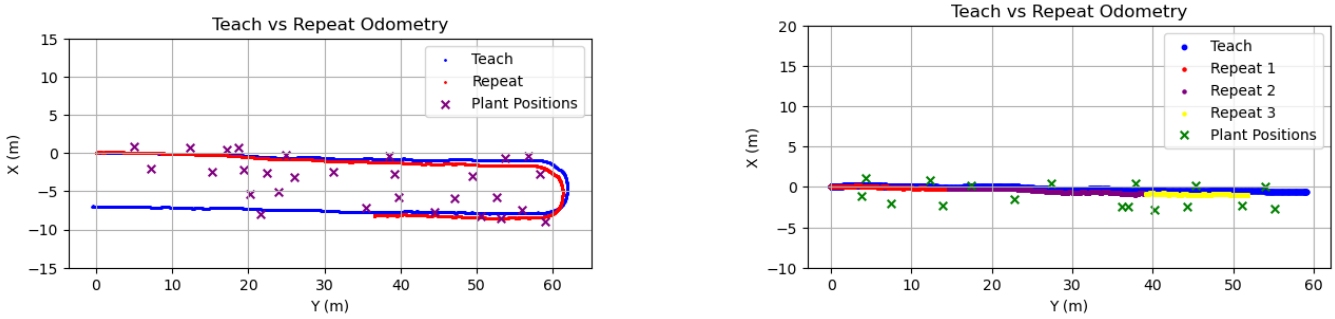
Data was acquired utilizing a front-facing Zed 2I stereo camera mounted on a wheeled robotic platform, specifically the Husky A200 developed by Clearpath Robotics [28]. The data acquisition was performed with a 2.1mm lens, capturing at a frame rate of 15 frames per second, in 720p resolution, with a field of view of 120 degrees. The Husky was maneuvered along two rows of the farm, which spans approximately 36,000 m^2 and includes 14 rows, each measuring approximately 60 meters in length. We captured data from two adjacent rows. Additionally, the Husky was outfitted with a Velodyne HDL-32E LiDAR scanner operating at 10Hz, with a known calibration to the camera. The LiDAR data was employed to generate a ground-truth point-cloud, facilitating the assessment of performance. Experiments were performed over two days, spaced a week apart. The first day's conditions were sunny without cloud cover, while the second day was characterized by overcast skies and light drizzle as shown in Figure 7. On each day, experiments were conducted at midday and during an afternoon session. The objective of these tests was to evaluate the capability of the system to repeat tasks under varied lighting conditions.

TABLE I: Examples of linguistic categories and corresponding instructions

Linguistic Category	Example Instruction
Direct scene graph concepts	Inspect the third plant in line 1.
Combination of scene graph concepts	Inspect plant 1 in line 1 and plant 5 in line 4
Same numerical iteration	Inspect plant 1 in line 1 and plant 1 in line 4
Co-reference resolution	Inspect the last plant in line 1.
Spatial relation	Inspect line 1.
Direct Scene Graph Concept with context	Carry berries to first plant in line 1
Failure	Jump into the sea.

B. Semantic Search

We carried out trials in delivering instructions, using the scene graph format described in Section IV. The various linguistic category instructions outlined in Table I were evaluated using three distinct LLMs. Initially, we selected linguistic categories to assess if the LLM could interpret straightforward scene graph concepts. We then proceeded to evaluate its ability to process combination instructions. Subsequently, the capacity of the LLM to accurately interpret



(a) Two row single instruction

(b) Single row multiple instructions

Fig. 8: Graphs showing odometry of teach passes and repeat passes with plants detected

TABLE II: Evaluation results: Number of correctly grounded commands per category. DSGC = Direct scene graph concepts, CSGC = Combination of scene graph concepts, SNI = Same numerical iteration, CR = Co-reference resolution, SR = Spatial Relation, DCWC = Direct Scene Graph Concept with context, FAIL = Failure

	GPT-4o-mini		Gemini 2 Flash Lite		Gemma 2-2b	
	Score	T(s)	Score	T(s)	Score	T(s)
DSGC	10	0.6	10	0.5	10	10
CSGC	10	1.16	10	0.5	6	15
SNI	10	1	10	0.6	4	15
CR	10	1	10	0.5	3	15
SR	10	1	10	0.9	3	45
DCWC	10	1	10	0.9	0	45
FAIL	10	0.8	10	0.5	10	45
Total	70 / 70	–	70 / 70	–	36 / 70	–

	SMD		SA		CMD		CA	
	1 Row	2 Row	1 Row	2 Row	1 Row	2 Row	1 Row	2 Row
SMD	S (0.13)	S (3.43)	S (0.32)	S (3.63)	S (0.53)	F	S (0.52)	F
SA	-	-	S (0.15)	S (3.02)	S (0.42)	F	S (0.53)	F
CMD	-	-	-	-	S (0.11)	S (3.42)	S (0.52)	F
CA	-	-	-	-	-	-	S (0.12)	S (3.23)

TABLE III: Evaluation Metrics: Success (S) or Failure (F) and RMSE (in brackets, meters) of navigation in different weather and time of day scenarios. SMD = Sunny Mid Day, SA = Sunny Afternoon, CMD = Cloudy Mid Day, CA = Cloudy Afternoon. The repeat passes were tested in one row and two-row paths.

scenarios where plants share enumeration but are located in different planting rows was examined. We further assessed co-referencing tasks involving identifiers like first, middle, and last, as well as more ambiguous tasks such as inspecting line 1. Furthermore we tested direct concepts but with context such as "Carry the berries to the first plant in line 1". Lastly, we tested instances of failure prompts, like "Jump into the sea". The chosen linguistic categories try to emulate typical instructions given to human workers within an agricultural environment. The output keyframe(s) were evaluated according to the scene graph passed as the input. GPT-4o-mini and Gemini 2 Flash Lite were called via API and Gemma 2-2b was run locally on board NViDIA Jetson Orin AGX. The zero-shot performance was tested without fine-tuning. Ten prompts were passed in each linguistic category and a point was awarded per correct keyframe answer. GPT-4o-mini and Gemini 2 Flash Lite returned perfect answers for every single prompt, with Gemini achieving better processing speed performance. Showing that the format of the scene graph output is able to sufficiently convey the structure of the scene graph, and the LLMs are able to interpret and ground the instruction in the scene graph. Gemma 2-2b performed worse compared to the other models in terms of both speed and score, and was unable to understand the structure of the scene graph and relate the semantic concepts.

C. Accuracy: Autonomy

To evaluate the accuracy of autonomy, we use the Gemini 2 Flash Lite model to send instructions to the robot. Varying instructions were given such that multi-row autonomy was tested (as in Figure 8a) and a single row with multiple stops was tested (Figure 8b). We generated four scene graphs, one from each of the weather and lighting conditions found in Table III. We conducted autonomy tests for each scene graph recorded in its current weather and lighting conditions and in each of the subsequent sessions. The robot repeats the path at 0.5m/s to keep the computational load manageable to allow the image data to be processed in time for responsive control. In Figure 7, the difference in the lighting and weather conditions can be seen. Afternoon sun creates more shadows than midday sun, while overcast conditions result in a varied sky with no shadows. Variation in lighting conditions strengthens the case for using Multi-Experience localisation. The results shown in Table III show that the scene graph through its robot layer is able to successfully localise and therefore this presented framework can realise the autonomy of ground robots in homogeneous agricultural environments. The failure cases reported were primarily due to lens flares from direct sunlight glare, which significantly limited visual localisation and subsequent autonomy. Figure 8 presents the teach paths in blue, for two-row and single-row runs. The sparse plant positions can be attributed to the performance of plant detection system from Osiris [3] used to produce the scene graph. Table III presents the Root Mean Square Error (RMSE) between the teach and repeat paths for each of the test cases. From the table we can see lower errors calculated for repeats conducted during the same conditions

as the teach and for the single row repeats compared to the two row repeats. From the trajectories, we can see that through the teach-and-repeat framework the robot was able to successfully navigate to the instructed locations and our system through its robots layer is able to allow for seamless interaction between classical robotic navigation pipelines and novel language models in homogenous agricultural environments.

VII. Conclusion

In this paper we present a framework that facilitates natural language instructions to be issued to a robot in an agricultural environment. We introduce the concept of a language-based 3D scene graph that supports autonomy via a robots layer. We utilize a Large Language Model that processes both a prompt and the produced 3D scene graph to create pertinent nodes for an autonomy planner from the input scene graph. We addressed the challenge posed by the uniformity of agricultural settings by leveraging the ordered sequencing and node labeling during the scene graph's creation. The autonomy planner produces a navigable plan based on the robots layer and passes this on to a controller for deployment on the robot. Our system was able to deal with simple and complex tasks successfully and returned correct keyframes in active agricultural environments. We also demonstrated that the robot was capable of autonomously navigating within a real farm environment, based on information captured during the generation of the scene graph and instructions from the LLM output, demonstrating the success of integrating a VTR framework with 3D Scene Graphs. By successfully showcasing the ability to take a natural language instruction and output an action on a robot, this work paves the way for unlocking user-friendly interfaces for language all around the world and advanced bespoke approaches for long-term monitoring of agricultural environments.

References

[1] M. O. Yerebakan and B. Hu, "Human–robot collaboration in modern agriculture: A review of the current research landscape," *Advanced Intelligent Systems*, vol. 6, no. 7, p. 2300823, 2024. 2

[2] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous robots*, vol. 42, pp. 957–975, 2018. 2

[3] A. Mukuddem and P. Amayo, "Osiris: Building hierarchical representations for agricultural environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 797–15 803. 2, 3, 4, 8

[4] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. D. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable task planning," *CoRR*, 2023. 2, 3

[5] Y. Liu, L. Palmieri, S. Koch, I. Georgievski, and M. Aiello, "Delta: Decomposed efficient long-term robot task planning using large language models," *arXiv preprint arXiv:2404.03275*, 2024. 2, 3

[6] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *IEEE Robotics and Automation Letters*, vol. 9, pp. 8921–8928, 2024. 2, 3

[7] A. S. Sahadevan, "Extraction of spatial-spectral homogeneous patches and fractional abundances for field-scale agriculture monitoring using airborne hyperspectral images," *Computers and Electronics in Agriculture*, vol. 188, p. 106325, 2021. 2, 3

[8] N. Hughes, Y. Chang, S. Hu, R. Talak, R. Abdulhai, J. Strader, and L. Carlone, "Foundations of spatial perception for robotics: Hierarchical representations and real-time systems," *arXiv preprint arXiv:2305.07154*, 2023. 3

[9] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A real-time spatial perception system for 3d scene graph construction and optimization," 2022. 3

[10] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From slam to spatial perception with 3d dynamic scene graphs," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021. 3

[11] J. Strader, N. Hughes, W. Chen, A. Speranzon, and L. Carlone, "Indoor and outdoor 3d scene graph generation via language-enabled spatial ontologies," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4886–4893, 2024. 3

[12] T. Silver, S. Dan, K. Srinivas, J. Tenenbaum, L. Kaelbling, and M. Katz, "Generalized planning in PDDL domains with pretrained large language models," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 3

[13] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. M. de Melo, J. B. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, "Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5021–5028. 3

[14] D. Shah, B. Osiński, S. Levine, and B. Ichter, "LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings, K. Chalupka, J. Medina, and Y. Zhu, Eds., vol. 205, 2023, pp. 492–504. 3

[15] Q. Xie, T. Zhang, K. Xu, M. Johnson-Roberson, and Y. Bisk, "Reasoning about the unseen for efficient outdoor object navigation," *CoRR*, vol. abs/2309.10103, 2023. 3

[16] M. Zuzuarregui and S. Carpin, "Leveraging llms for mission planning in precision agriculture," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025. 3

[17] J. Strader, A. Ray, J. Arkin, M. B. Peterson, Y. Chang, N. Hughes, C. Bradley, Y. X. Jia, C. Nieto-Granda, R. Talak, C. Fan, L. Carlone, J. P. How, and N. Roy, "Language-grounded hierarchical planning and execution with multi-robot 3d scene graphs," 2025. 3

[18] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, "Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9272–9279. 3

[19] M. V. Gasparino, V. A. Higuti, A. N. Sivakumar, A. E. Velasquez, M. Becker, and G. Chowdhary, "Cropnav: a framework for autonomous navigation in real farms," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 824–11 830. 4

[20] K. Kim, A. Deb, and D. J. Cappelleri, "P-agnav: Range view-based autonomous navigation system for cornfields," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 3366–3373, 2025. 4

[21] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of field robotics*, vol. 27, no. 5, pp. 534–560, 2010. 4, 7

[22] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013. 4

[23] C. Linegar, W. Churchill, and P. Newman, "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation," in *2015 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 90–97. 4

[24] M. Paton, K. MacTavish, M. Warren, and T. D. Barfoot, "Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat," in *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2016, pp. 1918–1925. 4, 7

[25] P. Amayo, P. Piniés, L. M. Paz, and P. Newman, "Geometric multi-model fitting with a convex relaxation algorithm," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8138–8146. 4

[26] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," in *ACM siggraph 2007 papers*, 2007, pp. 80–es. 5

[27] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, 2022. 5

[28] C. Robotics. (n.d.) Husky unmanned ground vehicle robot. Retrieved on 2024-03-04. [Online]. Available: https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/ 7