PyLoT Robotics 2026 Team Description Paper

Towa Yamashita Riku Kinoshita katasuya Honda
Koki Muramoto

November 30, 2025

Abstract. This paper introduces Runa, a mobile manipulator newly developed for RoboCup@Home 2026 by junior and high school students from Kaijo Junior and Senior High School PyLoT Robotics, and describes the software development and contributions made by PyLoT Robotics.

We have created our own affordable, production-ready and flexible robotic hardware. Furthermore, based on software using ROS2, we have focused on object recognition for object grasping, voice recognition and output, and task planning using LLM.

Using these, we aim to build a robot system that can handle more general-purpose tasks.

1 Introduction

PyLoT Robotics is a RoboCup team affiliated with Kaijo Junior and Senior High School. The team was founded in 2023 by high school students and all activities related to the team, including administration, development and outreach, are done by junior and senior high school students. The team has participated in regional RoboCup@Home leagues and is actively involved in educational activities for middle school students. We have continuously improved our software stack for perception, grasping, and planning.

This paper focuses on the progress of PyLoT Robotics's development for RoboCup@Home 2026, including spatial object detection using image processing and point cloud processing, improved robot arm control, and improved navigation accuracy for the 2026 competition.

We are also engaged in various activities to make robotics more accessible and to expand the robotics community middle and high school students.

2 Hardware

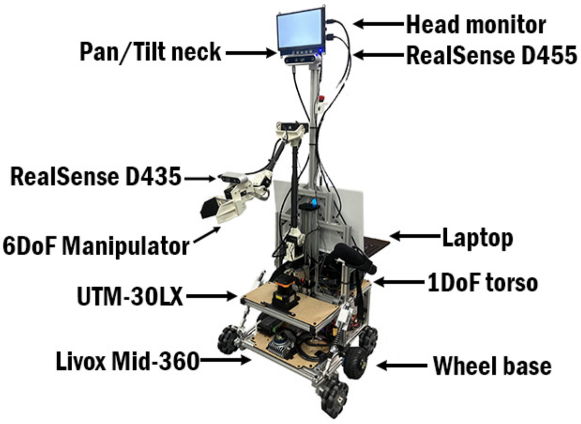


Fig. 1. Runa robot platform

PyLoT Robotics develops compact robots designed for high maintainability, as shown in the diagram.

The chassis utilizes a differential two-wheel drive system, while the main body features a disassemblable design using an aluminum frame and MDF, achieving light weight and low cost.

Additionally, 3D-printed parts are used for various connection points, ensuring excellent maintainability and easy component replacement.

The robot's underside houses a UTM-30LX for self-localization and navigation. The central section mounts a 6DOF robotic arm based on the Openmanipulator-x with modifications, along with various circuit boards, a small monitor, and an emergency stop button.

The upper section features an Intel RealSense RGBD camera and an orthogonal 2-axis servo motor, enabling a wide range of functions including object recognition and obstacle detection. A laptop PC is mounted at the rear center of the robot body, serving as the computational resource for controlling the robot.

3 Software

3.1 Software Overview

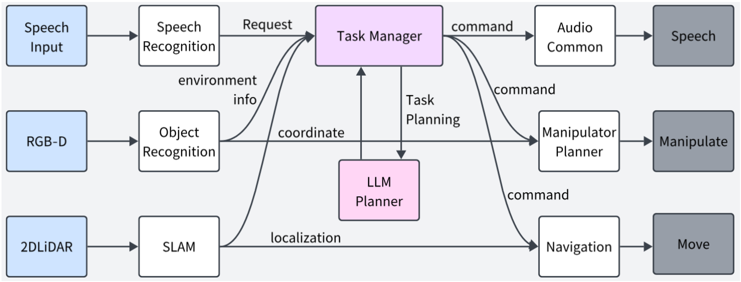


Fig. 2. System Overview

Our system integrates foundation models for end-to-end task execution. Speech input is transcribed by Whisper and processed by an LLM to extract user intent. For perception, we combine Detic-based full-scene segmentation with CLIP zero-shot classification, enabling recognition of unknown objects in RoboCup environments. Semantic information from these VLMs is fused into a map to support navigation and manipulation planning.

3.2 Speech Recognition

It uses Whisper[1] to convert speech into text. When you start a conversation, your spoken input is converted into text based on to pre-entered prompts, which are then fed into an LLM such as GPT4[2].

This enables the robot to simultaneously understand what to say from the user's input and what tasks to perform.

3.3 Object Detection

To address the unpredictability of RoboCup objects, we adopt a two-stage VLM pipeline: Detic provides open-vocabulary segmentation of all visible objects, and CLIP performs prompt-based zero-shot classification. Carefully designed prompts significantly improve recognition accuracy. The resulting semantic-labeled point cloud is used for downstream planning.

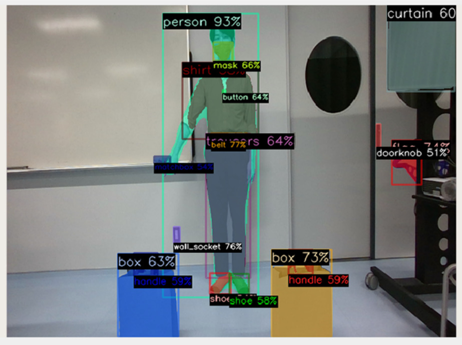


Fig. 3. Object Detection(Detic)

3.4 Grasping

We implemented a custom inverse-kinematics module for our robot arm. Target object coordinates obtained from semantic perception are transformed into the arm frame, and joint/cart trajectories are planned simultaneously while avoiding obstacles.

3.5 Task Planning

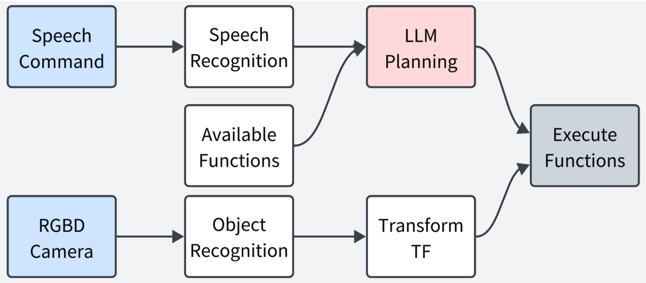


Fig. 4. Overview of a system that uses LLM to perform robot planning from speech

An LLM-based planner converts natural-language commands into executable action sequences. sing predefined action functions, object names, and environment information, the system generates structured plans capable of solving GPSR and EGPSR tasks.

3.6 Following a person

In our “Following a Person” module, we developed a navigation framework that combines A*-based path generation with nonlinear MPC for robust trajectory tracking. The A* planner provides collision-free global paths, while the nonlinear MPC controller ensures smooth and stable following in dynamic environments. This approach improved the robot’s ability to maintain distance, respond to human motion, and navigate safely in cluttered spaces. It performed strongly in the 2025 RoboCup@Home OPL Salvador, achieving one of high scores in the Help Me Carry task.

3.7 Active SLAM

We developed an Active SLAM system that performs SLAM and navigation simultaneously in previously unknown environments. By updating the map and planning motion in real time, the robot can localize accurately and generate reliable paths without prior environmental information using Slam-Toolbox[11]. This enables precise, adaptive movement and stable navigation even in dynamic or unstructured spaces. The proposed method achieved high scores at both the 2025 RoboCup@Home Japan Open and the 2025 RoboCup@Home OPL Salvador, demonstrating its effectiveness in real competition settings.

4 Our Research

4.1 Self Recovery Planning using VLM

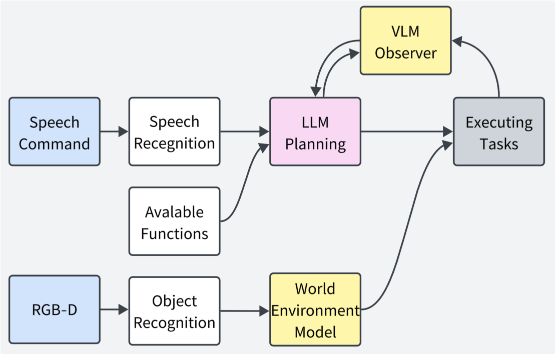
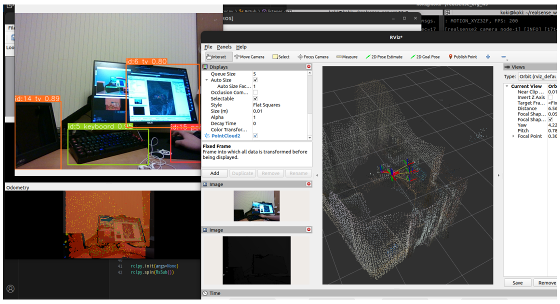


Fig. 5. Diagram of a robot planning system using VLM

We are working on the following research topic to improve the performance of the robot planning system. As mentioned above, we use an LLM for task planning, and we aim to further enhance planning by incorporating a VLM.

Based on the action plan generated by the LLM, the Task Executor executes the corresponding function. When a state is completed, the LLM queries the VLM about the robot's status, and the VLM checks whether the task was successfully completed using visual input from the robot's camera and provides feedback to the LLM. The LLM uses this information to perform replanning, enabling self-recovery. Since the VLM does not have any semantic information about the robot, it focuses on the target object and the target location, and generates prompts to obtain the status of the running function. A paper on this subject is currently under conference review.

4.2 Semantic Map using VLM



```
koki@koki: ~/realsense_w
: MOTION_XYZ32F, FPS: 200
[realsense2_camera_node-1] [INFO] [171
```

Queue Size	5
Auto Size	☑
Auto Size Fac...	1
Occlusion Com...	□
Selectable	☑
Style	Flat Squares
Size (m)	0.01
Alpha	1
Decay Time	0
Color Transfor...	
PointCloud2	☑

Fixed Frame
Frame into which all data is transformed before being displayed.

Type: Orbit (rviz_defau

Current View	Orb
Near Clip ...	0.01
Invert Z Axis	□
Target Fra...	<Fix
Distance	6.56
Focal Shap...	0.05
Focal Shap...	☑
Yaw	4.22
Pitch	0.78
Focal Point	0.30

```
41    rclpy.init(args=None)
42    rclpy.spin(RsSub())
```

Fig. 6. Semantic map from using VLM

Furthermore, we aim to construct a world environment model with semantic information so that robots can execute general tasks efficiently. The world model, which combines location and semantic information, is inspired by the clip-field[7] and can organize information in a cohesive space.

4.3 Adaptive Pan-Tilt Camera for Target Recognition

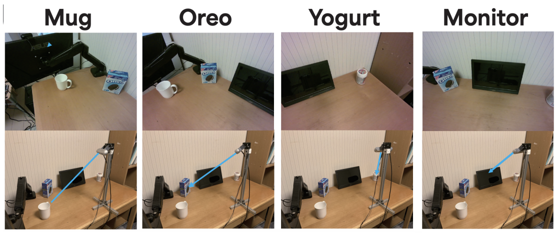


Fig. 7. View point controll for target recognition

There was an assumption that all the information necessary for self-healing was captured on the camera, but in this research, we eliminated that assumption and built a system that could collect the information necessary to complete a task on its own, assuming it would be installed on a mobile manipulator that required 360-degree monitoring.

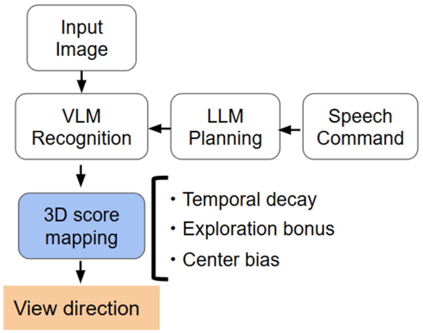


Fig. 8. View point controll overview

We have constructed a system similar to the one shown in **Fig.10** . Based on the input verbal instruction, the VLM determines whether the object is related to the instruction, and calculates the angle at which the camera should observe from the distribution of objects. The distribution of objects is set to decay over a certain period of time, and by inputting a gain in a direction that has not been observed for a certain period of time, it is possible to simultaneously perform a search operation.

5 Contribute

PyLoT Robotics aims to make robotics more accessible by hosting outreach events and providing step-by-step educational materials for younger students. We create an environment where middle school members can learn practical robot development early through structured courses with dedicated documentation and tools. Thanks to this program, two junior high school students joined our team and solved Receptionist task in the 2025 RoboCup Japan Open @Home Bridge Competition, demonstrating our impact as an educational robotics community.

6 Conclusions

This paper describes the robot platform, scientific contributions, and approach of PyLoT Robotics, a RoboCup team from Kaijo Junior And Senior High School, to participate in RoboCup 2026 at RoboCup@Home. By participating in RoboCup@Home, our team aims to develop autonomous mobile home service robots and make robotics more accessible to the public. In accordance with the principles of open source, we are publishing our various activities and results related to RoboCup. We plan to continue contributing to RoboCup by publishing our code and other materials even after the competition ends.

Acknowledgements

This paper is supported by Shimon Ajisaka, Dai Komukai, Yoshihiro Shibata and Takumi Nakamura.

References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In International Conference on Machine Learning, pages 28492–28518. PMLR, 2023.

[2] OpenAI. GPT-4 Technical Report. Technical report, 2023.

[3] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting Twenty-thousand Classes using Image-level Supervision. In ECCV, 2022.

[4] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In International conference on machine learning, pages 8748-8763. PMLR, 2021.

[5] Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics (Jan 2023), https://github.com/ultralytics/ultralytics

[6] Labbé, M., Michaud, F.: RTAB-Map as an open-source lidar and vi- sual simultaneous localization and mapping library for large-scale and long- term online operation. https://arxiv.org/pdf/2403.06341

[7] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory. In Robotics: Science and Systems, 2023.

[8] S. Macenski, T. Moore, DV Lu, A. Merzlyakov, M. Ferguson, From the desks of ROS maintainers: A survey of modern and capable mobile robotics algorithms in the robot operating system 2. https://github.com/ros-navigation/navigation2

[9] Steve Macenski and Matthew Booker and Josh Wallace. Open-Source, Cost-Aware Kinematically Feasible Planning for Mobile and Surface Robotics. https://arxiv.org/pdf/2401.13078

[10] Macenski, Steve and Martín, Francisco and White, Ruffin and Ginés Clavero, Jonatan. The Marathon 2: A Navigation System. https://github.com/ros-planning/navigation2

[11] Steve Macenski and Ivona Jambrecic. SLAM Toolbox: SLAM for the dynamic world. Journal of Open Source Software. https://doi.org/10.21105/joss.03085

Robot Runa Hardware Description

Robot Runa has the custom garbage collection mechanism. Specifications are as follows:

- Base: wheel base (differential pair), 2.5m/s max speed.
- Arm: 7DOF(1 DOF torso, 6DOF manipulator)
- Neck: 2DOF
- Head: Depth Camera, monitor display
- Robot dimensions: height: 1.4m (max), width: 0.6m depth 0.8m
- Robot weight: 30kg.
- RGB-D Sensors: Intel RealSense D435, Intel RealSense D455.
- LiDAR: Hokuyo UTM-30LX
- Microphones: CVM-V30PRO

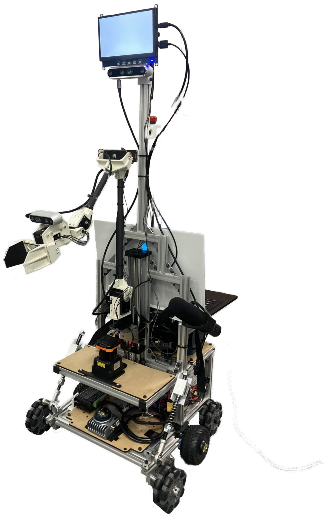


Fig. 9. Robot Runa

Robot's Software Description

For our robot we are using the following software:

- Platform: ROS2 Humble
- Navigation: Navigation2,EMCL
- Face recognition: Yolov8-Pose,Detic,GPT-4o
- Speech recognition: Whisper,Vosk.
- Speech generation: Audio-Common.
- Object recognition: Detic,CLIP,Yolov8
- Arms control: In-house arm motion planner

The following are the specifications of the laptop mounted on our Robot

- CPU: Ryzen7
- GPU: NVIDIA Geforce RTX 3050ti
- Memory: 16GB