EIC Robotics Team Description Paper

Nathampapop Jobsri, Tinapat Limsila, Thanakorn Sappakit, Suppakorn Boonprasert, Chayavich Asavakanoksilp, Tanakit Suetrong, Pattharaphol Chainiwattana, Korawish Thanasit, Patitta Ploypray, Pisit Pongsaran, Kittipong Sudjinda, Pongphol Suchirapatpong, Theetuch Chinachatchawarat, Thanasit Pakkaananchai, Nathamon Kongsawat, Bongkotmart Tiemmuang, and Naerunchara Prathumsuwan

Faculty of Engineering, Chulalongkorn University,
254 Phayathai Rd., Pathumwan, Bangkok 10330, Thailand
https://www.eng.chula.ac.th/en/

Abstract. Our main goal is to create a highly interactive robot as we believe it is the most important aspect if robots were to be integrated into our society. This paper will outline the changes done to our robot since RoboCup@Home 2022. We introduce a new wheel slip estimation method and more sensors to the robot which significantly impact its robustness and ability to localize. Computer Vision was one of our best performances during the competition. This year, we are testing new methods to automate the training process for the CV model. Subsequently, the success of CV has laid a foundation for a research paper to be published later in 2023. We hope that our robot will succeed in the upcoming RoboCup@Home 2023.

1 Introduction

Our team is from Chulalongkorn University, named Engineering Innovator Club or EIC. The teams founded by our club, Plasma-RX and Plasma-Z were the world champions in Robot Rescue League 2008 and Small-Size Soccer Robot League 2008, respectively[1]. Recently, we participated in RoboCup@Home 2022 open platform league and finished in second place. Since then, we have collaborated with RoboCup@Home Education to open a workshop for anyone interested in robotics, which is not exclusive to engineering students. The first workshop has achieved great success with participants from many universities in Thailand.

Despite our accomplishment in the recent competition, the robot did not perform to our expectations. The robot's performance was hindered by a lack of interactivity in both the environment and humans. We also encountered various hardware and software challenges, including an insufficient number of sensors for accurate localization and slow response times to human requests.

In this paper, we will address the challenges we faced during the previous competition and describe our solutions to these issues. We will also provide an

[1] http://robocupthailand.org/about.html

overview of the robot's navigation system, manipulation capabilities, computer vision, and natural language processing modules. By addressing these challenges, we aim to improve the performance of our robot and achieve a greater success in the upcoming RoboCup@Home 2023.

2 Navigation

2.1 Kinematics-based wheel slip estimation method

The kinematics-based wheel slip estimation technique is developed for the mobile planar platform. First, Wheel slip angles are calculated from kinematics-based analytical solutions of on-board measurable data which are the vehicle yaw rate, longitudinal and lateral accelerations, wheel rolling speeds, and steering angle[1]. Second, longitudinal and lateral velocities of the mobile base are derived from wheel slip angles, rolling speeds, and steering angles. Finally, through direct kinematic relation, all the state variables: traveling speed, sideslip, and radius of curvature can be obtained[2].

The preliminary experiment involves a 1:10 scaled vehicle with extremely random sideslip maneuver[1]. By comparing the result with the global positioning reference, the wheel slip angles could be well estimated despite the extreme slip. The method can be used to estimate wheel slip angles of any free-rolling wheels[2], given enough information. Thus, it can be used with many other wheeled mobile robots and mobile planar platforms.

The trajectory and radius of curvature of a novel wheelchair-exoskeleton hybrid robot using differential drive technique was also studied in different scenarios.[3]

2.2 Odometry sources

During the 2022 competition, our robot heavily relied on a single odometry source from a laser scanner. This is due to the resolution of our robot's encoder being too low and the IMU was not properly mounted, thus causing multiple slips. As a result, the decision was made to lower the weight of both sensors as they were deemed not as reliable as intended at the time.

To solve the issue mentioned above, this year, the encoder resolution is increased to 4096 ppr from 90 ppr and the IMU is enclosed in a vibration-dampening case. Additionally, two optical flow sensors have been added as extra odometry sources. An extended Kalman filter is responsible for the fusion of all odometry data.

2.3 SLAM

2D SLAM has proven to be insufficient in the competition, especially in a semi-outdoor environment(restaurant) where there is a lot of hard to track objects

for a Lidar. Thus, we will be utilising Spatio Temporal Voxel Layer or `STVL`[2] for tracking obstacles in 3D. Also we are testing a full 3D SLAM approach using `RTAB-Map`[3] which will be used if a situation allows. Additionally, our robot now supports real time mapping, which means it does not need any pre-recorded occupancy grid map to navigate. As a result, tasks that require navigation in an unknown area, such as the restaurant, can now be performed.

For tasks that requires human robot interaction, such as person following, we will implement the last observed position technique(LOP) to deal with the case that the person is instantaneously turning around the corner [4].

3 Manipulation

Our grasp generator from the previous year involved having the robot to capture the scene, and then pass an image to a computer vision model for a bounding box of a desired object. The position of the object was calculated using a single point depth at the middle of bounding box. This method has a major flaw. Each object has a different shape, size, and weight that should be considered when calculating for grasp motion. However, the single point depth is incapable of obtaining those data; thus, a different approach is needed.

In order to identify the shape and size of the object, we implement `OctoMap`[4] using Point Cloud from the depth camera. We also use a `RANSAC algorithm`[5](Point Set Generation Network for 3D Object Reconstruction from a Single Image) to fill in the empty Point Cloud. Once the object is identified, our grasp generator algorithm based on the study of MoveIt! Grasp and MoveIt! Deep Grasp is used to produce grasp pose according to the object shape and size. Additionally, the implementation of the `OctoMap` allows the robot to recognize obstacles in its planned trajectory, which enable the arm to pick up the object without running into collision objects.

For robot's end effector, Inverse kinematics of planar manipulator and custom design of a gripper for specific task (eg. liquid handling) were presented[5]. Parallel elastic actuation[6] can be applied for reducing energy consumption in manipulation tasks with asymmetric load requirement in anti-gravity direction.

4 Natural Language Processing (NLP)

4.1 Changes

NLP went through a major overhaul. We changed from Azure Cognitive Service[6] to a fully offline stack. We found that an offline stack delivers a more consistent

[2] `https://github.com/SteveMacenski/spatio_temporal_voxel_layer`
[3] `http://introlab.github.io/rtabmap/`
[4] `http://octomap.github.io/`
[5] `https://github.com/leomariga/pyRANSAC-3D`
[6] `https://azure.microsoft.com/en-us/products/cognitive-services`

response, as certain places have less-than-ideal internet connections. One example is Azure TTS would take 6 or more seconds to return the speech file which breaks the interactivity. Another advantage is the voice recording and analysis are not being collected by online services for increased privacy.

4.2 High-Level Architecture Overview

The robot listens for the wake word if required. After receiving the wake-word, Automatic Speech Recognition (ASR) will be activated. The ASR then transforms the audio input into text. The text is then parsed into Natural Language Understanding (NLU); in this process, the text is passed into an intent classification model and an entity extractor which returns valuable intents, and entities (object, name, location). The intent and entities are delivered to the Dialogue Manager. It calculates the action and tracks the conversation by analyzing the current intent and the previous action to maintain the flow of the conversation. It is also responsible for sending the intent and entities to other components such as to the Natural Language Generation (NLG) system for generating a text response for ROS to control the robot or to third-party applications. Text-to-speech (TTS) is required to convert the generated sentences into natural-sounding speech responses. Finally, HTTPS is used to communicate between the core components of NLP. By combining HTTPS and GPU acceleration, the result is a very fast request-to-response NLP, further reaching the goal of a highly interactive robot.

To sums up, our voice assistant consists of 4 important components: Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Natural Language Generation (NLG), and Text-to-Speech (TTS).

4.3 Natural Language Processing Components

The stack consists of PicoVoice Porcupine[7] for the wake-word, OpenAI Whisper[8] for Speech-to-Text, Rasa Open Source[9] for natural language understanding (NLU & NLG), and Coqui[10] for Text-to-Speech (TTS).

The Porcupine PicoVoice is an offline model that's easy to train using the Porcupine console. "Hey Walkie" is the wake word. From testing, the wake is extremely responsive to various accents or other ways of pronouncing "Hey Walkie."

OpenAI Whisper, an ASR model trained on more than 680,000 hours of multi-lingual speech, is an offline model with high accuracy in its transcription. Our test showed the Whisper model outperforms the Azure Cognitive Service in terms of accuracy.

[7] `https://picovoice.ai/platform/porcupine/`
[8] `https://openai.com/blog/whisper/`
[9] `https://rasa.com/docs/rasa/`
[10] `https://github.com/coqui-ai`

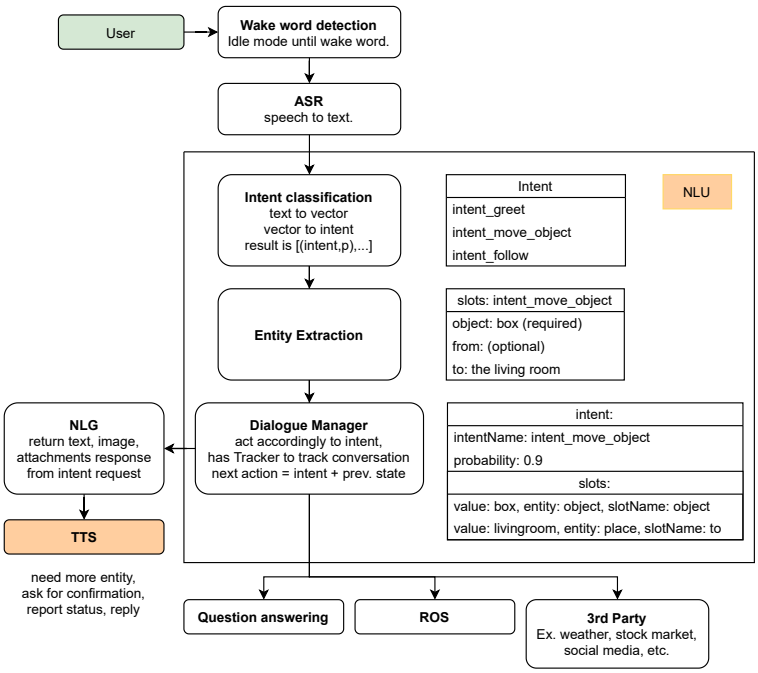

Fig. 1. NLP architecture diagram

Rasa will be used as the machine learning framework to automate text and voice-based conversations (NLU & NLG). To deliver voices near human quality, our focus has been improving the flexibility of user input. Every individual phrasing of a sentence varies from one to another.

Coqui TTS will play an important role in this process. It is an open-source library for advanced Text-to-Speech generation[11].

5 Computer Vision

5.1 Virtual environment model training

For computer vision, we spent lots of time during the competition gathering, labeling, and verifying the dataset for the training of the model. There has been discussion for creating a photorealistic virtual environment for model training; this is one of the possibilities of streamlining the overall. Last year, our computer vision model performed exceptionally well; however, a significant amount of time was spent preparing images for model training. This involved manually processing and labeling thousands of images, which was a time-consuming and labor-intensive task. To improve this process, we are implementing automation

[11] `https://tts.readthedocs.io/en/latest/`

for image labeling for the upcoming competition. This allows us to prepare our training data efficiently.

To automate the labeling process, we are using a virtual environment that gives us full control over the angle. We are using Unreal Engine 5 to export labeled coordinates. Images from the camera are stitched together using Neutral Radiance Field (NeRF) to create a photorealistic 3d model. This approach will allow us to quickly and accurately generate large amounts of labeled training data for our computer vision model. Additionally, this technology enables anyone to easily generate a large amount of hyper-realistic datasets of their choice, with a quality comparable to or occasionally even superior to datasets created manually from captured images.

5.2 Instance Segmentation

YOLOv8[12] is used for object detection and instance segmentation of the robot. It is a cutting-edge, state-of-the-art model that can accurately detect objects in real time. It is an anchor-free model that reduces the number of box predictions, which results in faster non-maximum suppression and predictions than previous YOLO models. It can also output the masks and bounding boxes for each object, making manipulation tasks much easier to implement. Finally, when tested on a custom dataset, YOLOv8 is more accurate and 35% faster than our previous year's model, YOLOv5.

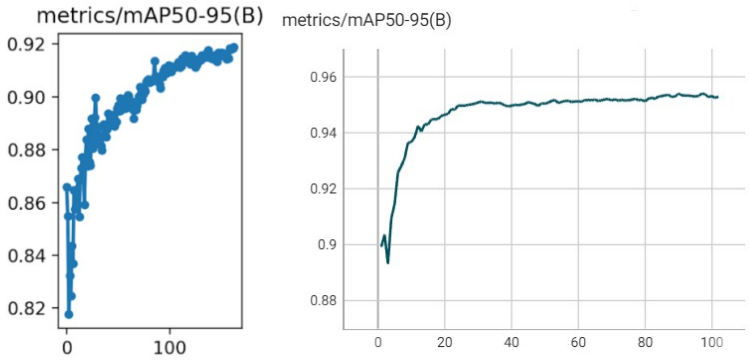

Fig. 2. YOLOv5 (left) and YOLOv8 (right) mAP Comparison

5.3 Machine learning-based garbage sorting

The success of the CV has led to the creation of a research paper that will be published later this year. The paper presents a proof-of-concept for a machine learning-based garbage sorting system. The system was trained on a small

[12] `https://github.com/ultralytics/ultralytics`

dataset of 90 images, which were fed into the YOLOv5 model. The results showed an accuracy of 93.3% in the specific circumstances where the background was fixed and the garbage had a consistent shape and appearance. However, this may not accurately reflect real-world conditions. The paper also discusses benchmarking to determine the ideal number of epochs for achieving the highest accuracy with minimal underfitting and overfitting.

5.4 Human Facial Recognition

The model we use is `face_recogntion`[13] by Adam Geitgey. It was built using dlib's state-of-the-art face recognition built with deep learning, which has a 99.38% accuracy on the Labeled Faces in the Wild benchmark. The algorithm finds and identifies the bounding boxes of all the faces.

5.5 Pose Estimation

OpenPose[a] is used for human posture estimation. It is a multi-person posture prediction capable of detecting joints. The advantages of using OpenPose compares to Mediapipe from last year are the following

- Supports multiple instances of humans in a frame without losing performance (FPS).
- Higher joint estimation accuracy, especially when multiple people are partially overlapped.

[a] `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

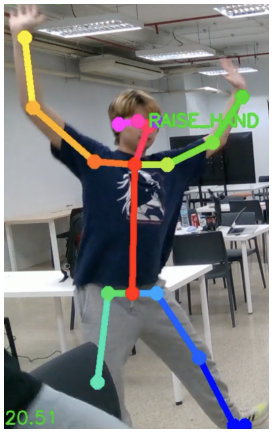

Fig.3. OpenPose Demo

6 Conclusion

This paper discusses the improvement and new features of the robot based on the challenges we faced in RoboCup@Home 2022. For Navigation, we come up with kinematics-based wheel slip estimation that can be adapted to any mobile planar platform. Many sensors are added for additional odometry. Additionally, the grasp generator for Manipulation is improved so that it can categorize objects and generate grasp motion accordingly. For NLP, the stack was changed to an offline model with better interactivity and responsiveness. CV has added new features such as YOLOv8 instance segmentation, and OpenPose joint detection that allows the robot to better interact with its surroundings. Moreover, the research paper about machine learning-based garbage sorting, inspired by the previous competition will soon be published. With our issues being rectified, we believe that our robot will achieve greater success in RoboCup@Home 2023.

[13] `https://github.com/ageitgey/face_recognition`

References

[1] R. Chaichaowarat and W. Wannasuphoprasit, "Kinematics-based analytical solution for wheel slip angle estimation of a rwd vehicle with drift," *Engineering Journal*, vol. 20, no. 2, pp. 89–107, May 2016. DOI: 10.4186/ej.2016.20.2.89.

[2] R. Chaichaowarat and W. Wannasuphoprasit, "Wheel slip angle estimation of a planar mobile platform," *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pp. 163–166, 2019. DOI: 10.1109/ICA-SYMP.2019.8646198.

[3] R. Chaichaowarat, S. Prakthong, and S. Thitipankul, "Transformable wheelchair–exoskeleton hybrid robot for assisting human locomotion," *Robotics*, vol. 12, no. 1, p. 16, Jan. 2023. DOI: 10.3390/robotics12010016.

[4] R. Algabri and M.-T. Choi, "Deep-learning-based indoor human following of mobile robot using color feature," *Sensors*, vol. 20, no. 9, 2020, ISSN: 1424-8220. DOI: 10.3390/s20092699. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2699.

[5] R. Chaichaowarat, A. Sirichatchaikul, W. Iamkaew, and N. Phondee, "Affordable pipetting robot: Gripper design for automatic changing of micropipette and liquid volume control," pp. 1275–1280, 2022. DOI: 10.1109/AIM52237.2022.9863351.

[6] R. Chaichaowarat, J. Kinugawa, A. Seino, and K. Kosuge, "A spring-embedded planetary-geared parallel elastic actuator," pp. 952–959, 2020. DOI: 10.1109/AIM43001.2020.9158998.

Walkie Hardware Description

- Base: Custom aluminum profile with acrylic casing
- Driver: 2 Wheels differential drive
- Manipulator: DOBOT CR3, 6DOF
- Elevator: 1 DOF
- End effector: TPU-Flexible adaptive gripper
- Head: The bird, Pan-tilt unit using 2 servo actuators
- RGB-D sensor: Intel Realsense D415
- Stereo sensor: Stereolab ZED2
- LIDAR sensor: Hokuyo Ust-10lx
- Battery: 1x 24V 100Ah
- Computer: Acer Nitro5, Core i7, RTX 3070
- Robot dimensions: Width = 0.55 m, Length = 0.55 m, Height = 0.8 m
- Robot Weight: 80 kg

Fig.4. Walkie robot

Robot's Software Description

- OS: Ubuntu 20.04 LTS
- Middleware: ROS Noetic
- Navigation: `move_base` ROS and slam-toolbox
- Manipulation: MoveIt!, OMPL Library
- Computer Vision:
  - Instance Segmentation: YOLOv8
  - Human Facial Recognition: `face_recognition`
  - Pose Estimation: OpenPose
- Natural Language Processing:
  - Automatic Speech Recognition: Rhasspy & OpenAI Whisper
  - Natural Language Understanding: Rasa
  - Text-to-Speech (TTS): Coqui TTS