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 ambition is to create a service robot that is as interactive as possible. We have updated our robot base on the experience and issues gathered from RoboCup@Home 2022. This year, there is a significant improvement to the localization of the robot. We developed a kinematics-based wheel slip estimation method that requires only on-board measurement data. This technique can be applied for many wheeled mobile robots and automobile. Moreover, we switch from Microsoft Azure to the open-source OpenAI Whisper for natural language processing (NLP), due to its high-accuracy offline characteristics. computer vision (CV) was one of our strong suits, achieving a 94.5% accuracy. However, the process of training the model was tedious, requiring the labeling of 43,000 images. This year, we will automate the labeling process with the use of a virtual environment. By capturing the "task" environment, we can label thousands of images in minutes. The success in CV has inspired a research paper on machine learning-powered garbage sorting machine. By using NLP and CV to improve the user experience, our service robot can provide a less obstructive environment for people to do what they do best. We hope that our service robot will success 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 2008 in Robot Rescue League and Small-Size Soccer Robot League, respectively[1]. Recently, we have participated in RoboCup@Home 2022 open platform league and finished in second place.

Despite our accomplishment in the recent competition, the robot could not perform up to our expectations. The robot's performance was hindered by a lack of interactivity with both the environment and human. We also encountered

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

various hardware and software challenges, including an insufficient number of sensors for accurate localization and slow response times to human requests. Additionally, the object detection model training was inefficient.

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 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 greater success in the upcoming RoboCup@Home 2023 competitions.

2 Navigation

2.1 Kinematics-based wheel slip estimation method

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

The preliminary experiment involves 1:10 scaled vehicle with extremely random sideslip maneuver[1]. By comparing the result with 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 wheeled mobile robot and mobile planar platforms.

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 an extra odometry source. An extended Kalman filter is responsible for the fusion of all odometry data.

2.3 SLAM

2D-SLAM has proven to be insufficient during our time in the competition, especially in a semi outdoor to outdoor environment where there is a lot of direct

sunlight and transparent objects such as glass. Thus, we will be utilizing `RTAB-Map`[2] for 3D SLAM with stereo depth camera. The map is created by merging the scanned data from laser range finders and stereo camera which is projected onto the horizontal plane [3].

Since our robot has to be autonomous, the goal point cannot be set manually, so the input from RGBD camera is required. When the state machine reaches the state that needs input from the Computer Vision module, the Computer Vision module has to return the x, y, and z position of the goal point. Then, the 3D coordinate, in pixels, is transformed into a real world coordinate, in metres, related to the camera coordinate. The position of the detected object is mapped. The robot can set the goal point and calculate the best path by A* algorithm. As for the task that require human 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 algorithm of grasp generator from the previous year involved having the robot take a photo using its camera, and then passing that photo through a computer vision model to label and identify the bounding box of our preferred object. The distance between the robot and the object was determined using Point Cloud in the bounding box's center, and a grasp was generated using the arm's motion that was previously calculated. To control the arm, Moveit! was used; however, this could cause a few issues. First, there might be a discrepancy in the object's position. Second, each object had a different shape, size, and weight, requiring a different algorithm and robot arm position. In order to identify the shape of the object and virtually simulate the arm motion, we implement `OctoMap` [3] along with Point Cloud from the camera to solve those problems. The initial value-set data from the iPhone's lidar sensor is used to aid in our calculations. We also use a Point Set Generation Network for 3D Object Reconstruction from a Single Image or `RANSAC algorithm` [4] to fill in the empty Point Cloud. Later, we generate grasp by using MoveIt! Grasp or MoveIt! Deep grasp to categorize the object and the algorithm to grabbed it. Last but not least, we give our robot the ability to recognize the furniture nearby, such as the table and the cabinet, enabling the arm to pick up the object without running into them.

4 Natural Language Processing (NLP)

4.1 Discussion

Last year, the natural language processing (NLP) component of our robot was the most disappointing aspect of its performance. We experienced issues with

[2] `http://introlab.github.io/rtabmap/`
[3] `http://octomap.github.io/`
[4] `https://github.com/leomariga/pyRANSAC-3D`

the need to use Microsoft's always-online Azure Cognitive Service, which was unreliable due to unstable internet connectivity. This year, we are moving toward a fully offline model for speech-to-text processing. In our proof-of-concept testing, the offline model OpenAI's Whisper had good accuracy and performed similarly or better than Azure in some cases. Another issue with our previous NLP model was its inflexibility in accepting inputs, as sentences needed to match predetermined patterns. We are addressing this issue by implementing a more versatile and adaptable model for natural language processing. Overall, these improvements will enhance the interactivity and responsiveness of our robot to human requests. Finally, an offline NLP model can be extremely beneficial in numerous ways, including preventing network latency and protecting our privacy and data.

4.2 High-Level Architecture Diagram

Firstly, when the robot receives a wake word, a voice assistant will be activated. Secondly, Automatic Speech Recognition (ASR) transforms audio input into text. Then, the text is then parsed into Natural Language Understanding (NLU) to transform human language into a machine-readable format. In this process, the text is put in an intent classification model which converts the text to vectors and classifies the intent. Next, the text is parsed into an entity extractor to get the value of variables required by each intent.

The intent and entities are delivered to the Dialogue Manager. It calculates the action and tracks the conservation 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, to ROS for controlling the robot, and to third-party applications. Finally, Text-to-speech (TTS) is required to convert generated sentences into natural-sounding speech responses.

Therefore, A voice assistant consists of 4 important tools: Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), Natural Language Generation (NLG), and Text-to-Speech (TTS).

4.3 Automatic Speech Recognition (ASR)

Rhasspy [5] will be used as a voice assistant toolkit that can support technologies for important tasks such as wake word detection, speech-to-text, and Natural Language Understanding (NLU). Supporting technologies that will be used in Rhasspy are Porcupine[6] and DeepSpeech [5].

- Porcupine is a high-performance offline system that listens for a wake word to activate a voice assistant.

[5] `https://github.com/rhasspy/rhasspy`
[6] `https://github.com/Picovoice/porcupine`

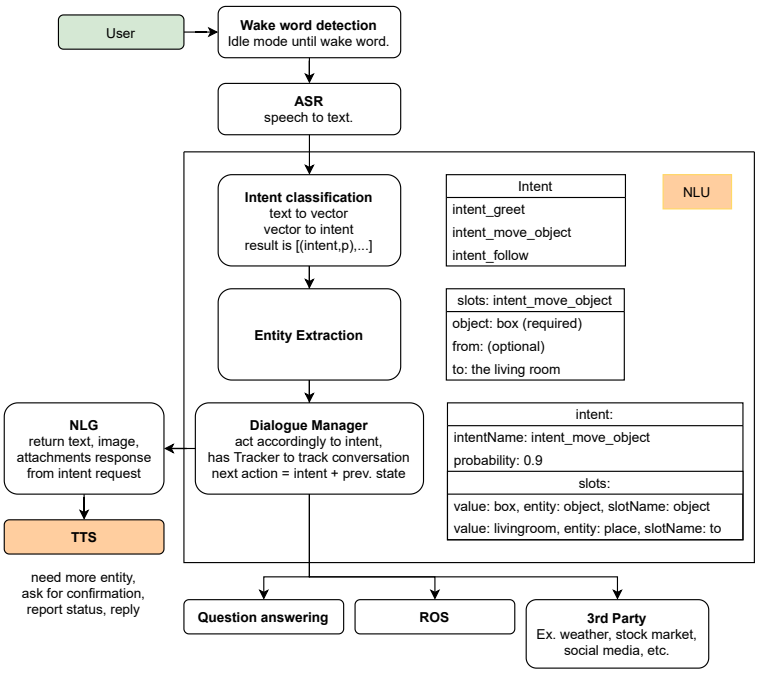

Fig. 1. NLP architecture diagram

– Whisper[7] transforms audio input from humans to text. Whisper is an automatic speech recognition (ASR) system trained on 680,000 hours of multilingual languages, English being one of the languages. The model is trained by non-profit organization OpenAI. The code for operation and training is open source which makes implementation easier.

4.4 Natural Language Understanding

Rasa[8] will be used as a machine learning framework to automate text and voice-based conversations. It will also be added to the Rhasspy toolkit. Moreover, the framework offers easy integration, highly detailed documentation, and tools for testing and data labeling.

4.5 Text-to-Speech (TTS)

To deliver voices near human quality, Coqui TTS will play an important role in this process. It is an open-source library for advanced Text-to-Speech generation[9]. It was not only designed to achieve the best trade-off among ease-of-

[7] https://github.com/openai/whisper
[8] https://rasa.com/docs/rasa/
[9] https://tts.readthedocs.io/en/latest/

training, speed, and quality but also built on the latest research such as Aligner Network and FastSpeech2. Coqui TTS also provides many alternative high-performance Deep Learning models in both Text2Speech and vocoder models which allow us to select the best models based on our situation.

5 Computer Vision

5.1 Virtual environment model training

For computer vision, the Bulk of the time during the competition was 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 for image labeling for the upcoming competition. This will allow us to prepare our training data for training for 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 label 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 Object Detection

You Only Look Once Version 5 (YOLOv5[10]), released by Ultralytics in June 2020, is implemented in the object-detecting part of the robot. It is a cutting-edge convolutional neural network (CNN) that accurately detects objects in real-time. This method processes the entire image using a single neural network, then divides it into parts and predicts bounding boxes and probabilities for each component. These bounding boxes are weighted by the expected probability. Detected products are then delivered after non-max suppression. In YOLOv5, the CSP (Cross Stage Partial Networks) is utilized as a backbone to extract valuable properties from an input image. The object detection accuracy of YOLOv5 is comparable to that of YOLOv4, but it is 88% smaller and 180% faster.

[10] `https://github.com/ultralytics/yolov5`

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 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`[11] by Adam Geitgey. It was built using dlib's state-of-the-art face recognition built with deep learning, which evaluated at 99.38% accuracy on the Labeled Faces in the Wild benchmark. The algorithm finds bounding boxes of all the faces and identifies them.

5.5 Person Tracking

For the person tracking of the robot, `yolov4-deepsort`[12] is used. The YOLOv4 algorithm performs object detection using deep convolutional neural networks. The module uses YOLOv4 output to detect objects and feeds these items into Deep SORT to produce a very precise object tracker.

5.6 Pose Estimation

For the pose estimation, MediaPipe[13] is used. We utilise MediaPipe Hands and MediaPipe Pose to process the data from the color image.

- MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to extrapolate 21 3D hand landmarks from a single frame. We use this model to calculate and track extreme precision hand movement and position, such as the angle of the hand, the angle of the finger, and the direction the finger is pointing.
- MediaPipe Pose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation masks on the whole body from RGB video frames. The real-time pose of a person is estimated using this model. It can determine whether someone is standing or sitting, or whether they are raising their hand or not.

[11] `https://github.com/ageitgey/face_recognition`
[12] `https://github.com/theAIGuysCode/yolov4-deepsort`
[13] `https://mediapipe.dev/`

6 Conclusion

This paper discusses the detail of the problems during RoboCup@Home 2022, and our solution to said problem. For Navigation, we have come up with kinematics-based wheel slip estimation that can be used on any mobile planar platform. Many sensors are added for additional odometry. Additionally, the grasp generator for Manipulation is improved so that it can categorized objects and generate grasp motion accordingly. For NLP, a speech-to-text processing is changed to an offline and more adaptive model for a better interactivity and responsiveness. For CV, a method of model training and labelling using virtual environments is tested to produce training data quickly and accurately for computer vision. Moreover, the research paper about machine learning based garbage sorting, inspired by the previous competition will soon be published. We hope that our method will help our robot to be more interactive to its surroundings and achieve greater success in the RoboCup@Home 2023.

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] A. S. Ali Yeon, K. Kamarudin, R. Visvanathan, *et al.*, "Feasibility analysis of 2d-slam using combination of kinect and laser scanner," *Jurnal Teknologi*, vol. 76, Oct. 2015. DOI: 10.11113/jt.v76.5858.

[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] A. Y. Hannun, C. Case, J. Casper, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: 1412.5567. [Online]. Available: http://arxiv.org/abs/1412.5567.

Walkie2 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 = 0.63 m, Length = 0.64 m, Height = 1.35 m
- Robot Weight: 80 kg

Fig.2. Walkie2 robot

Robot's Software Description

- OS: Ubuntu 20.04 LTS
- Middleware: ROS Noetic
- Navigation: `move_base` ROS Package and RTAB-Map
- Manipulation: MoveIt!, OMPL Library
- Computer Vision:
  - Object Detection: YOLOv5
  - Human Facial Recognition: `face_recognition`
  - Person Tracking: `yolov4-deepsort`
  - Pose Estimation: MediaPipe
- Natural Language Processing:
  - Automatic Speech Recognition: Rhasspy & OpenAI Whisper
  - Natural Language Understanding: Rasa
  - Text-to-Speech (TTS): Coqui TTS