Hibikino-Musashi@Home 2024 Team Description Paper

Kosei Isomoto, Akinobu Mizutani, Fumiya Matsuzaki, Hikaru Sato, Ikuya Matsumoto, Kosei Yamao, Takuya Kawabata, Tomoya Shiba, Yuga Yano, Atsuki Yokota, Daiju Kanaoka, Hiromasa Yamaguchi, Kazuya Murai, Kim Minje, Lu Shen, Mayo Suzuka, Moeno Anraku, Naoki Yamaguchi, Satsuki Fujimatsu, Shoshi Tokuno, Tadataka Mizo, Tomoaki Fujino, Yuuki Nakadera, Yuka Shishido, Yusuke Nakaoka, Yuichiro Tanaka, Takashi Morie, and Hakaru Tamukoh

Kyushu Institute of Technology
The University of Kitakyushu
hma@brain.kyutech.ac.jp
https://www.brain.kyutech.ac.jp/~hma/

Abstract. This paper provides an overview of the techniques employed by Hibikino-Musashi@Home, which intends to participate in the domestic standard platform league. The team has developed a dataset generator for training a robot vision system and an open-source development environment running on a Human Support Robot simulator. The large language model powered task planner selects appropriate primitive skills to perform the task requested by users. The team aims to design a home service robot that can assist humans in their homes and continuously attends competitions to evaluate and improve the developed system.

1 Introduction

Hibikino-Musashi@Home (HMA) is a robot development team comprising students at the Kyushu Institute of Technology and the University of Kitakyushu in Japan. The team was founded in 2010 and has participated in the RoboCup@Home JapanOpen in the open platform league (OPL), domestic standard platform league (DSPL), and Simulation-DSPL. It has recurrently participated in the RoboCup@Home league since 2017 and will participate in RoboCup 2024 to present the outcomes of their latest developments and research. In addition to the RoboCup, the team participated in the World Robot Challenge (WRC) 2018 and 2020 as well as in the service robotics category of the partner robot challenge (real space). HMA focuses on the development of robot vision systems, particularly dataset generation systems for the training of object-recognition systems. It also develops libraries for primitive tasks, including object recognition, grasping point estimation, and navigation. Task planning is their latest topic of interest, which uses a large language model (LLM) to plan a task by selecting primitive tasks in a dynamic environment.

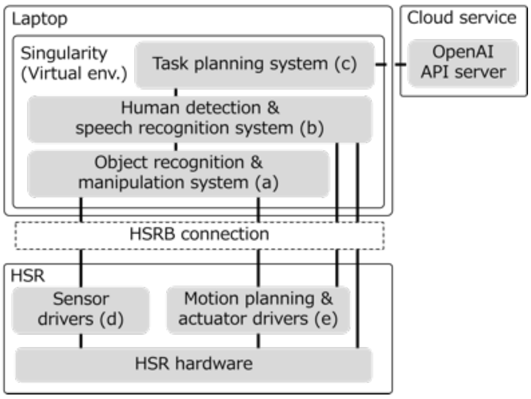

Laptop
Cloud service
Singularity (Virtual env.)
Task planning system (c)
OpenAI API server
Human detection & speech recognition system (b)
Object recognition & manipulation system (a)
HSRB connection
HSR
Sensor drivers (d)
Motion planning & actuator drivers (e)
HSR hardware

Fig. 1. Block diagram providing an overview of the HSR system. [HSR, Human Support Robot; ROS, robot operating system]

2 System Overview

We use an external laptop (ThinkPad X1 Extreme Gen5) mounted on the back of the TOYOTA Human Support Robot (HSR) to compute systems require high computational power. The computer inside the HSR is also used to run basic HSR libraries, such as sensor drivers, motion planners, and actuator drivers. Most of the system is complete within the mounted laptop and internal HSR computer except the LLM API server.

Figure 1 presents an overview of the HMA's software systems for HSR [20]. Our system is on Singularity [5], a virtual environment tool built on a laptop mounted on the HSR. The system comprises an object recognition and manipulation system (a), a human and speech recognition system (b), and a task planning system (c), which are all connected to the sensor and actuator drivers (d) and (e) on the HSR's internal computer, respectively. The task planning system (c) communicates with the OpenAI API server [3] via the Internet.

3 Perception

3.1 Object Recognition

The object-recognition system is a crucial component of robot systems. We have adopted a strategy that leverages both the YOLOv8 [9] and Language Segment-Anything [2]. We select the best system for each task depending on the target objects.

YOLO YOLOv8 has low latency with accuracy that is sufficient to recognize the known objects in the competition. We fine-tune the YOLO system to recognize

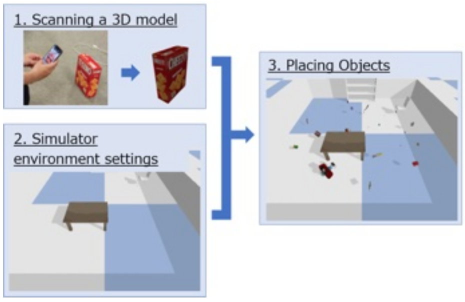


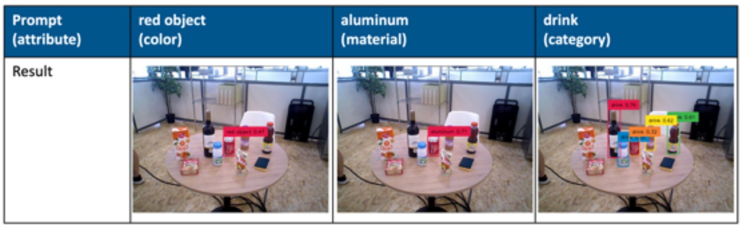
Fig. 2. Placing scanned objects on a 3D simulator.

the objects used in the competition using enormous training dataset generated by the 3D simulation system based on the PyBullet [7] simulator [16]. As shown in Fig. 2, to generate the training dataset, first, we create a 3D model of each object by using a smartphone with LiDAR sensors. The scanned 3D objects are spawned in the 3D environment, and the objects as well as the environment are shot from various angles to create a dataset with 500,000 images in under 2 h using a six-core CPU simultaneously. The light conditions, placement of the furniture, and texture of the background (floor, wall, and ceiling) are changed to randomize a domain at each shot. The annotation data for the training data can be generated automatically by the system; no annotation process by humans is required.

Language Segment-Anything Language Segment-Anything [2] is a combination of Grounding DINO [13] and Segment Anything [11]. Grounding DINO is a recognition system that can be tuned by providing text prompts. Segment anything model (SAM) can also output the segmentation mask of the object without any fine-tuning. In this method, the human has to provide the text prompt and the recognition accuracy depends on this prompt. However, the system can be tuned quickly without fine-tuning; thus, the known objects available on the competition site can be recognized without extensive preparation. The target object can be detected by giving different types of prompts such as color, material and category as shown in Fig. 3. Humans are required to choose and tune the prompt by checking the recognition result in advance.

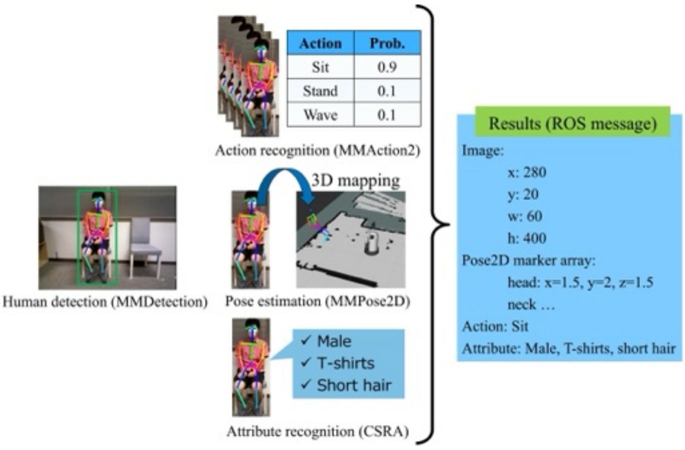
3.2 Speech Recognition

We have built a real-time speech recognition system. HSR acquires the speaker's voice in frames and transmits it sequentially to an external laptop computer. Our system uses Voice Activity Detection (VAD) [19] to determine when to start and stop recording. Post recording, our system automatically starts speech recognition using only the speech section's audio data. We use Whisper [17], a high-accuracy speech recognition method.



Prompt (attribute)	red object (color)	aluminum (material)	drink (category)
Result			

Fig. 3. Example of the prompts for Language Segment-Anything.



Action recognition (MMAction2)
3D mapping
Human detection (MMDetection)
Pose estimation (MMPose2D)
✓ Male
✓ T-shirts
✓ Short hair
Attribute recognition (CSRA)

Action	Prob.
Sit	0.9
Stand	0.1
Wave	0.1

Results (ROS message)

```
Image:
      x: 280
      y: 20
      w: 60
      h: 400
Pose2D marker array:
      head: x=1.5, y=2, z=1.5
      neck …
Action: Sit
Attribute: Male, T-shirts, short hair
```

Fig. 4. Placing scanned objects on a 3D simulator.

3.3 Human Detection

We use the MM Libraries provided by OpenMMLab [4] for human recognition. Figure 4 demonstrates our human recognition system. We use MMDetection to detect humans and obtain cropped images. We detect human key points through MMPose and recognize human action using MMAction2. In addition, we use Class-Specific Residual Attention (CSRA) [21] to obtain human attributes, such as gender, hairstyle, and clothing.

3.4 Human Tracking

We use YOLOv7 [18] and StrongSORT [8] for human tracking tasks such as *Carry My Luggage* and *General Purpose Service Robot (GPSR)*. In these tasks, the robot must to follow the human to the destination in a crowded environment.

Humans can be detected by YOLOv7; however, it cannot identify a human uniquely. To track a particular human, we use StrongSORT, a multi-object tracking(MOT) system.

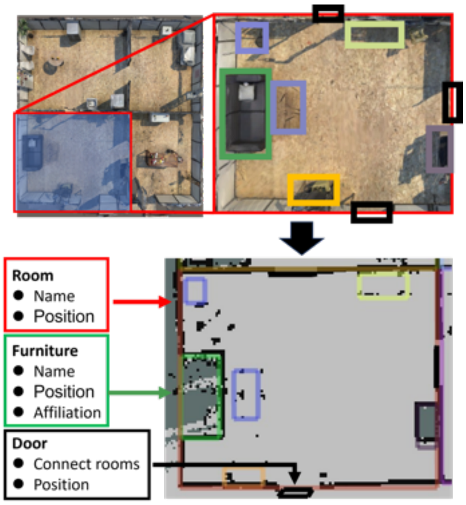


Fig. 5. Example of a semantic map

3.5 Semantic Map

A semantic map is required for the autonomous task execution of the service robots. As shown in Fig. 5, we add the semantic information about the room, furniture, and door to a pre-acquired environment map created using Real-Time Appearance-Based Mapping (RTAB-map) [12]. Each room and furniture has a label with the name and location information using an array of two-dimensional (2D) coordinates representing the vertex information of the contours. Moreover, the furniture has information regarding the room in which it is located, and each door has information regarding the room to which it is connected. Before the task execution, we define areas of the room and the positions of the door and furniture to input to a configuration file by humans. A robot utilizes that information to not only determine the semantic location of the robot or humans in an arena but also plan the path to the target position.

The path planning algorithm using semantic maps works as follows. First, the robot receives the 2D coordinates of the destination location. Then, it determines the destination room and the room of the robot's current position using the inside/outside determination method based on the outer product. Finally, it calculates the shortest path from the robot's current position to the destination. This calculation includes the Euclidean distance from the robot's position to the door, between doors that the robot passes through to reach the desired room, and from the door to the specified furniture. The algorithm can change the path by setting the door as passable. This allows the robot to replan its path by setting the door as impassable after detecting that it is closed.

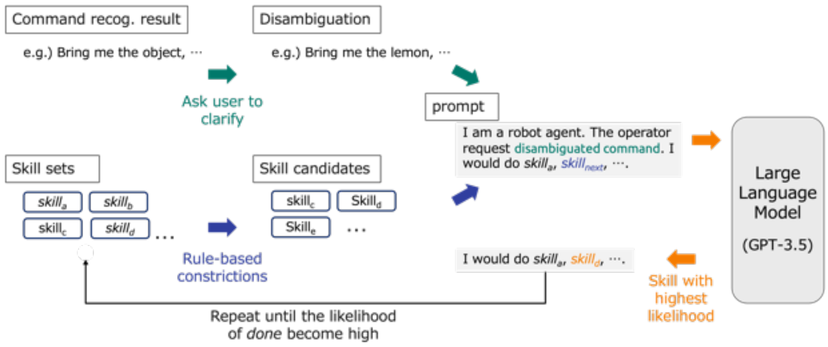

Command recog. result
e.g.) Bring me the object, ⋯
Ask user to clarify
Disambiguation
e.g.) Bring me the lemon, ⋯
prompt
I am a robot agent. The operator request disambiguated command. I would do $skill_a$, $skill_{next}$, ⋯.
Large Language Model
(GPT-3.5)
Skill sets
$skill_a$
$skill_b$
skill$_c$
$skill_d$
⋯
Rule-based constrictions
Skill candidates
skill$_c$
Skill$_d$
Skill$_e$
⋯
I would do $skill_a$, $skill_d$, ⋯.
Skill with highest likelihood
Repeat until the likelihood of *done* become high

Fig. 6. Overview of the task planning

4 Task Planning

A task planning system is required to accomplish various requests from a user in real-world environments. As shown in Fig. 6, the proposed system plans the task using command recognition results and skill sets based on the SayCan system [6].

The voice requests from the users are converted into text. The given command may have an ambiguous word, for example, *Bring me the object*. In this case, the system asks the user for the proper noun of the ambiguous word/phrase. Then, based on the user's answer, the system replaces the ambiguous word with the proper noun. The skill set is the set of available skills for the robot. The possible skills are selected based on the rule-based constrictions, for example, the same skill is not repeated consecutively, and `grasp` is after `find_obj`. The likelihood of each possible skill is calculated by the LLM, and the skill with the highest likelihood is selected as the next robot's task until the likelihood of *done* becomes the highest.

To plan the task, the system requires common and environment-specific knowledge. For example, to bring a bottle of water to the family, the possible location can be thought of using common knowledge and the actual location depends on each home environment. To acquire the environment-specific information, we have proposed a brain-inspired memory acquisition model with hippocampus functions [14,15]. This model is designed to be implemented on low-power consumption hardware such as FPGAs and dedicated chips [10]. We think this model is necessary for the robot to work in a home environment and the latest research activity will be demonstrated in the *Open Challenge*.

5 Reusability

We have published our development workspace used in RoboCup 2021 on the open-source HSR simulators [1] on GitHub[1]. It includes documentation and sample programs with motion-synthesis and object-recognition libraries. This simulator workspace enables us to develop the robot system even without the physical HSR. It can also be used for the testing and evaluation of the robot system. Currently, we are working on the development of an open-source development workspace for physical HSR with the virtual environment on Singularity.

6 Conclusions

This paper describes the techniques for creating an intelligence system for home-service robots. The automatic dataset generation system is essential to train the visual system of a service robot in a limited time. The task planning system is sufficiently powerful to create the robot's action by the human's spoken request, and necessary primitive tasks are continuously developed by the team to enhance the functions of the home service robot.

Acknowledgment

This paper is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This paper is also supported by Joint Graduate School Intelligent Car, Robotics & AI, Kyushu Institute of Technology student project, YASKAWA electric corporation project, JSPS KAKENHI grant number 22K17968, and JST SPRING grant number JPMJSP2154.

References

1. hsrb_robocup_dspl_docker, `https://github.com/hsr-project/hsrb_robocup_dspl_docker`, (Accessed 2023-10-27)
2. Language Segment-Anything, `https://github.com/luca-medeiros/lang-segment-anything`, (Accessed 2023-10-27)
3. OpenAI API server, `https://platform.openai.com/`, (Accessed 2023-10-24)
4. Openmmlab, `https://github.com/open-mmlab`, (Accessed 2023-10-24)
5. Singularity, `https://apptainer.org/docs/`, (Accessed 2023-10-24)
6. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al.: Do as I can, not as I say: Grounding language in robotic affordances (2022). https://doi.org/10.48550/arXiv.2204.01691
7. Coumans, E., Bai, Y.: Pybullet, a python module for physics simulation for games, robotics and machine learning (2016)

[1] `hma_wrs_sim_ws` (`https://github.com/Hibikino-Musashi-Home/hma_wrs_sim_ws`)

8. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia pp. 1–14 (2023). https://doi.org/10.1109/TMM.2023.3240881
9. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics (Jan 2023), `https://github.com/ultralytics/ultralytics`
10. Kawashima, I., Tateno, K., Morie, T., Tamukoh, H.: A memory-based entorhinal-hippocampal model and its fpga implementation by on-chip rams. In: IEEE International Symposium on Circuit and Systems (ISCAS2022)
11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023). https://doi.org/10.48550/arXiv.2304.02643
12. Labbé, M., Michaud, F.: RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. Journal of Field Robotics **36**(2), 416–446 (2019). https://doi.org/10.1002/rob.21831
13. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection (2023). https://doi.org/10.48550/arXiv.2303.05499
14. Mizutani, A., Tanaka, Y., Tamukoh, H., Katori, Y., Tateno, K., Morie, T.: Brain-inspired neural network navigation system with hippocampus, prefrontal cortex, and amygdala functions. In: 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) (Nov 2021). https://doi.org/10.1109/ISPACS51563.2021.9651058
15. Mizutani, A., Tanaka, Y., Tamukoh, H., Tateno, K., Nomura, O., Morie, T.: A knowledge acquisition system with a large language model and a hippocampus model for home service robots. In: IEICE Tech. Rep. vol. 123, pp. 13–18
16. Ono, T., Kanaoka, D., Shiba, T., Tokuno, S., Yano, Y., Mizutani, A., Matsumoto, I., Amano, H., Tamukoh, H.: Solution of world robot challenge 2020 partner robot challenge (real space) **36**(17–18), 870–889 (2022). https://doi.org/10.1080/01691864.2022.2115315
17. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. https://doi.org/10.48550/arXiv.2212.04356
18. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475 (2023)
19. Wiseman, J.: py-webrtcvad. `https://github.com/wiseman/py-webrtcvad`, (Accessed 2023-10-24)
20. Yamamoto, T., Terada, K., Ochiai, A., Saito, F., Asahara, Y., Murase, K.: Development of human support robot as the research platform of a domestic mobile manipulator. ROBOMECH Journal **6**(1) (2019). https://doi.org/10.1186/s40648-019-0132-3
21. Zhu, K., Wu, J.: Residual attention: A simple but effective method for multi-label recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 184–193 (October 2021)

Appendix 1: Robot's Software Description

The following is the software stack of our robot system, as shown in Fig. 7.

- OS: Ubuntu 20.04
- Middleware: ROS Noetic
- State management: SMACH (ROS)
- Speech recognition: Whisper [17]
- Object detection: YOLO and Language Segment-Anything [9,2]
- Human detection/action recognition:
  - MM Libraries (MMDetection, MMPose, MMAction2) [4]
  - YOLO and StrongSORT [18,8]
- Attribute recognition: CSRA [21]
- SLAM: rtabmap [12] (ROS)
- Path planning: move_base (ROS)

The following are the specifications of the laptop mounted on our HSR.

- Model name: ThinkPad X1 Extreme Gen5
- CPU: Intel Core i9-12900H
- RAM: 32GB
- GPU: NVIDIA GeForce RTX 3080Ti (16GB)

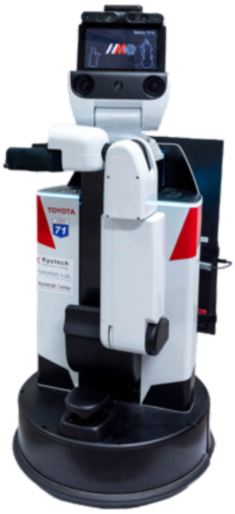


Fig. 7. HSR

Appendix 2: Competition results

Table 1 shows the results achieved by our team in the recent competitions. We have been participating in the RoboCup and World Robot Challenge for several years. Our team has won several prizes and academic awards.

Appendix 3: Links

- Team Video
  `https://youtu.be/VKKz-PcQsvc`
- Team Website
  `https://www.brain.kyutech.ac.jp/~hma`
- GitHub
  `https://github.com/Hibikino-Musashi-Home`
- Facebook
  `https://www.facebook.com/HibikinoMusashiAthome`
- YouTube
  `https://www.youtube.com/@hma_wakamatsu`

Table 1. Results of the recent competitions. [DSPL, domestic standard-platform league; JSAI, Japanese Society for Artificial Intelligence; METI, Ministry of Economy, Trade and Industry (Japan); OPL, open-platform league; RSJ, Robotics Society of Japan]

Competition	Result
RoboCup 2017 Nagoya	**@Home DSPL 1st** @Home OPL 5th
RoboCup JapanOpen 2018 Ogaki	@Home DSPL 2nd **@Home OPL 1st** JSAI Award
RoboCup 2018 Montreal	**@Home DSPL 1st** P&G Dishwasher Challenge Award
World Robot Challenge 2018	**Service Robotics Category** **Partner Robot Challenge Real Space 1st** METI Minister's Award, RSJ Special Award
RoboCup 2019 Sydney	@Home DSPL 3rd
RoboCup JapanOpen 2019 Nagaoka	**@Home DSPL 1st** **@Home OPL 1st**
RoboCup JapanOpen 2020	@Home Simulation Technical Challenge 2nd **@Home DSPL 1st** @Home DSPL Technical Challenge 2nd **@Home OPL 1st** **@Home OPL Technical Challenge 1st** @Home Simulation DSPL 2nd
RoboCup Worldwide 2021	@Home DSPL 2nd **@Home Best Open Challenge Award 1st** **@Home Best Test Performance: Go, Get It! 1st** **@Home Best Go, Get It! 1st**
World Robot Challenge 2020	**Service Robotics Category** **Partner Robot Challenge Real Space 1st**
RoboCup Asia-Pacific 2021 Aichi Japan	**@Home DSPL 1st** **@Home OPL 1st**
RoboCup JapanOpen 2021	**@Home DSPL 1st** **@Home DSPL Technical Challenge 1st** @Home OPL 2nd **@Home OPL Technical Challenge 1st**
RoboCup 2022 Bangkok	@Home DSPL 3rd @Home Best Open Challenge Award @Home Robo-host (Party-Host highest score in Stage I tasks)
RoboCup JapanOpen 2022 Tokyo	**@Home DSPL 1st** @Home DSPL Technical Challenge 2nd @Home OPL 2nd **@Home OPL Technical Challenge 1st**
RoboCup JapanOpen 2023 Shiga	@Home DSPL 3rd **@Home DSPL Open Challenge 1st** @Home OPL 2nd **@Home OPL Open Challenge 1st**
RoboCup 2023 Bordeaux	@Home DSPL 2nd