UT Austin Villa@Home
2024 Team Description Paper

Yuqian Jiang Chang Shi Steven D Patrick Justin Hart
Luis Sentis Peter Stone

February 12, 2024

Abstract. UT Austin Villa has participated in six RoboCup@Home competitions, performing respectably in each. What is more exciting, however, is that we have begun a strong program of research that has been in part inspired by our efforts in this competition. It is our intention to build a comprehensive service robot system which is used in our laboratories, in real-world deployments, and to compete in RoboCup@Home. In this Team Description Paper, you will find the highlights of our efforts and our plans for 2024.

1 Introduction

Using the RoboCup@Home team as a focal point for inter-department and inter-laboratory collaboration, UT Austin Villa@Home has pursued an ambitious research program towards the goal of the development of a comprehensive service robot system. We want to enter RoboCup@Home not with a suite of different programs for each round, but with a single program which is capable of competing and winning.

UT Austin Villa@Home is a collaborative effort between PIs and students in the Computer Science, Mechanical Engineering and Aerospace Engineering departments at the University of Texas at Austin, with a diverse set of research interests driving our team. We have competed in seven RoboCup@Home events. In 2007, we took second place. In 2017, we entered into the newly-formed Domestic Standard Platform League (DSPL) and took third place, having received our robot only a couple of months before the competition. In 2018, the team developed a design intended to allow us to develop a single system which would enter into all of the stages of the competition, encompassing knowledge representation, mapping, and architectural aspects. The team advanced to the second stage and was able to score in difficult tasks such as Enhanced General Purpose Service Robot (EGPSR). In 2019, we improved the system with better perception and manipulation modules. In 2021, we continued to develop

our object recognition and manipulation capabilities using the HSR simulator, and finished in the 3rd place in the 2021 competition. In 2022, we continued to strengthen our perception pipeline and re-designed the person tracking module, and qualified for the second stage in Bangkok. In 2023, we explored methods to combine LLMs with task and motion planning for interactive mobile manipulation. While we were unable to demonstrate our capabilities fully at the 2023 competition due to hardware issues, the progress made would set a good starting point for 2024 and open many research opportunities. Our efforts have resulted in seven publications [1,2,3,4,5,6,7], with more in progress. Going into 2024, we plan to further improve the core components of our system and develop more rigorous approaches to the tasks. We will focus on leveraging state-of-the-art robot foundation models for perception, planning, and human-robot interaction.

2 Software and Scientific Contributions

This section describes the component technologies we developed across multiple tasks for our robot architecture, knowledge representation, semantic perception, object manipulation, and person following on top of the HSR software stack. To the extent possible, we built our approach in a manner consistent with our ongoing Building-Wide Intelligence project [8]. While using a different hardware platform, many of the objectives and capabilities are the same. Indeed we have previously designed an underlying architecture that is common to the two platforms [6].

2.1 Robot Architecture

Our architecture is designed for service robots to handle dynamic interactions with humans in complex environments. The three-layer architecture, as shown in Figure 1, outlines integration of the robot's skill components, such as perception and manipulation, with high-level reactive and deliberative controls. The top layer sequences and executes skills, and is reactive during execution to respond to changes. A central knowledge base facilitates knowledge sharing from all the components. The deliberative control layer uses the knowledge base to reason about the environment, and can be invoked to plan for tasks that cannot be statically decomposed. Details on implementation of these layers can be found in our recent paper [6].

2.2 Knowledge Representation and Planning

Our knowledge representation subsystem stores grounded robot knowledge in a SQL database in order to allow for fast access and easy querying. For instance, in the GPSR task of the 2023 competition, the knowledge base is used to store object categories and their corresponding locations. Fig. 2 shows the knowledge base after the robot has detected a ketchup bottle on the dining table. Queries can be formed using custom C++ and Python libraries. The knowledge base can

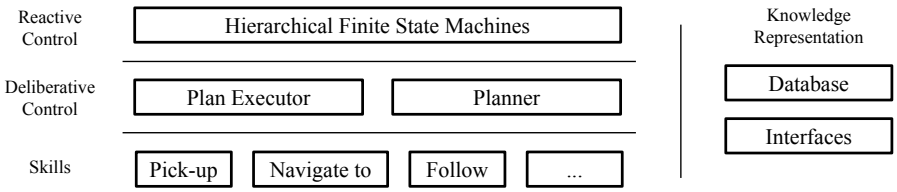

Fig. 1. Implementation of our robot architecture on HSR.

be interfaced through a simple predicate logic form which can be then imported for task planning. Our task planning module utilizes Answer Set Programming (ASP) to describe the rules for planning and reasoning, and the solver Clingo to generate optimal task plans. Core to our KR subsystem is the ability to reason about hypothetical objects. This task planning module is crucial to our solution of GPSR and EGPSR tasks. Details on our knowledge representation and planning system can be found in our paper [2]. We plan to improve and integrate this system with our recent work on LLMs and task planning [7] to enable natural language plan queries.

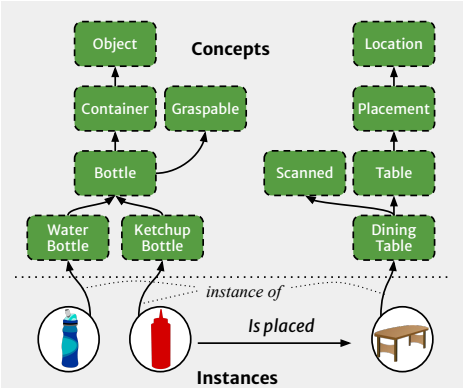

Fig. 2. Visualization of a knowledge base grounded in the robot's perception.

2.3 Perception

We employ a semantic perception module whose purpose is to process raw video and depth data from the robot's sensors and extract information that can be processed by the manipulation, navigation, and knowledge reasoning modules. The main output representations are a query-able point cloud of objects in the environment and a partial 3D map of the world.

The main input to our semantic perception module is RGBD camera data. Compressed RGB and depth images from the robot are streamed to an offboard computer that runs the perceptual system. This image data is then consumed by finding objects via the YOLO object detection network [9]. We have trained YOLOv5 models and built them in the TensorRT library for high performance inference. The processing time of one frame is only a few milliseconds on the backpack laptop. Next, semantic information about the world is synthesized in two main ways: a partial 3D environmental map and object cloud. For the former, regions of the point cloud corresponding to detected objects are fused together over time in a probabilistic Octree representation based on Octomap [10], which allows for the realtime construction of a partial 3D map of the world. For the latter, point estimates of the locations of objects are stored in a KD-Tree and wrapped with an efficient querying interface that integrates with our knowledge representation system. The synthesized semantic information is then made available to plugins in an event-based model, where a plugin can request access to semantic information that it wants to operate on. Plugins used include custom RANSAC edge detectors used to detect surfaces, and bounding box fitting on the 3D map for use in manipulation.

A significant limitation is the partial nature of the 3D environmental map. Only a partial map is constructed due to the realtime processing constraint; namely, full views of the world cannot be stitched together at framerate using the Octomap technology. Alternatively, GPU-based techniques for combining full point clouds could potentially overcome this limitation, and thus provides a direction for future development. Benefits of having full 3D environmental maps include the ability to directly localize objects with respect to the robot. In 2024, we plan to improve our semantic perception framework with state-of-the-art methods to generate open-vocabulary 3D scene graphs. Specifically, we will use vision-language models to obtain object and relation descriptors instead of a closed set of YOLO labels. This improvement will enable our system to handle unknown objects and open-vocabulary queries.

2.4 Manipulation

The purpose of our manipulation system is to enable the pick up and put down of diverse objects of different shapes and sizes. Our manipulation stack consists of three main components which we describe below: grasp pose generation, parallel motion planning, and closed-loop correction.

First, our semantic perception system provides 3D bounding boxes for objects worth manipulating. Based on these bounding boxes, potential grasp poses are computed that place the gripper on the top of the object as well as on all sides, with multiple possible rotations of the wrist. Of these poses, invalid configurations are filtered out by projecting the gripper onto the object and seeing if there is a collision.

Once grasp poses are determined, motion plans need to be determined in order for the robot to achieve a desired grasp pose. In order to do this quickly, we employ a parallel motion planning architecture built on top of the Moveit

framework [11]. Our motion planning architecture is comprised of primary and secondary nodes. The secondary nodes handle generating motion plans for each potential grasp pose, while the primary node coordinates and handles executing motion plans. Specifically, secondary nodes plan in parallel, and the first motion plan found is what is executed. The rationale behind this is that different grasp poses will require different yet unknown amounts of time for finding motion plans. Since motion planning takes a significant amount of time, reducing this bottleneck greatly speeds up the entire manipulation pipeline. Furthermore, the Moveit framework can sometimes crash when trying to find plans. In our setup, this problem is mitigated: If a secondary node terminates from such a crash, then the other secondary nodes are still present, allowing the system to continue functioning.

Next, executing a motion plan precisely is usually not feasible. This is because, as the plan is executed, the software solely uses odometry to control its position and the resultant drift can cause errors in how much the robot thinks it has moved. To overcome this obstacle, we slightly modify desired grasp poses by having the gripper be some offset away from the object. This way, after a motion plan is generated and executed, the robot's gripper is close to the object, but there remains a small gap. We take advantage of this small gap by employing a real-time, closed-loop grasp adjustment based on the fast YOLO detections applied to images from the HSR's hand camera. We use the position of the generated 2D bounding box to align the gripper with the target object. A proportional controller is used to publish a velocity command to the robot base based on the distance between the center of the hand camera image and the center of the bounding box. This practically means that the robot shifts slightly to align the gripper perfectly with the centroid of the object. The gap is then closed by moving in a straight line towards the object, leading to a successful grasp.

2.5 Person Following

To achieve robust and efficient person-following capabilities, perception, robot gaze control, and navigation must be effectively integrated. Recently, vision-based human recognition has dramatically improved with new software that relies on deep learning-based technologies, but these approaches have a limited range of sight. To resolve this problem, laser-based methods [12][13] and various sensor fusion techniques combining face recognition and leg detection have been employed [14][15]. However, there remain major difficulties include handling occlusions, identifying target people among crowds, and effectively detecting human faces. To surpass these limitations, new techniques have been devised that rely on extra features, such as the detection of clothes, bags, and shoes [16].

Another problem is due to the use of passive perception techniques where the robot stays stationary, thus losing its target. Therefore, it is highly desirable for robots to achieve active perception such that people can be followed despite their movement. Many researchers have studied this problem within the topic of active perception or visual sensor planning [17]. This kind of problem is usually

intractable because there are too many variables. However, using prior knowledge, context, and logical assumptions about the environment, it is possible to find solution approximations. If a robot is aware of the connectivity between spaces, when the target suddenly disappears from the robot's view, one strategy could be to navigate to the anticipated point using the last observed location to look for the target. This space connectivity can be simplified with the use of a topology map or graph . One other key factor is that robot skills should be integrated in harmony with the perceptual processes to improve a robot's ability to adapt to the various dynamic circumstances. For example, actions such as searching for a target, tracking, and navigating should be properly coordinated. To achieve such coordination, we employ the behavior-tree method [18] to sequence skills.

In summary, we develop person following capabilities using sensor fusion, active search using trajectory and waypoints predictions, and construct fully autonomous behaviors to follow people including temporary losses of the target being followed. Details on our person following approach can be found in our recent paper [3].

In 2022, we started experimenting with recent deep learning methods for robust multi-object tracking. We trained a person re-identification model from a large dataset of labeled person images using triplet loss. We also incorporated state-of-the-art tracking algorithms on the MOT Challenge [19] such as BoT-SORT [20]. In 2024, we plan to further improve our person tracking system for interactive tasks such as *Carry My Luggage* and *Receptionist*.

2.6 Object Coreference Through Pointing

The ability to interpret point gestures enables natural human-robot interaction in *Carry My Luggage* and *Hand Me That* (discontinued in 2023). Our solution to this problem leverages MediaPipe [21], running on RGB-D image data obtained by the Xtion sensor on the HSR's head. Identification of a point gesture starts by using either MediaPipe Pose to track the endpoints of the arms (locating the hands) or the MediaPipe Palm Detection Model to track the palms directly in the color image. The region determined to contain each palm is then run through the MediaPipe Hand Landmark Model, and the landmarks are then turned into 3D landmarks using the registered depth map from the Xtion's depth sensor. From these and a ray running from the base through the tip of the index finger is computed. The distance of the centroid of each tracked object from the Semantic Perception Module (Section 2.3) is then compared against the computed ray, and the object that is closest to the ray is determined to be the object indicated by the point gesture.

3 Conclusion

UT Austin Villa@Home has been a strong competitor and has a tradition of synergistic research our RoboCup@Home team and our other research efforts.

RoboCup@Home has become a driving force in robotics research at UT Austin. We look forward to seeing everyone in Eindhoven in summer 2024.

References

1. Rishi Shah, Yuqian Jiang, Haresh Karnan, Gilberto Briscoe-Martinez, Dominick Mulder, Ryan Gupta, Rachel Schlossman, Marika Murphy, Justin Hart, Luis Sentis, and Peter Stone. Solving service robot tasks: Ut austin villa@home 2019 team report. In *AAAI Fall Symposium on Artificial Intelligence and Human-Robot Interaction for Service Robots in Human Environments (AI-HRI 2019)*, November 2019.
2. Yuqian Jiang, Nick Walker, Justin Hart, and Peter Stone. Open-world reasoning for service robots. In *Proceedings of the 29th International Conference on Automated Planning and Scheduling (ICAPS 2019)*, July 2019.
3. Minkyu Kim, Miguel Arduengo, Nick Walker, Yuqian Jiang, Justin W Hart, Peter Stone, and Luis Sentis. An architecture for person-following using active target search. *arXiv e-prints*, pages arXiv–1809, 2018.
4. Justin W. Hart, Rishi Shah, Sean Kirmani, Nick Walker, Kathryn Baldauf, Nathan John, and Peter Stone. Prism: Pose registration for integrated semantic mapping. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2018.
5. Justin Hart, Harel Yedidsion, Yuqian Jiang, Nick Walker, Rishi Shah, Jesse Thomason, Aishwarya Padmakumar, Rolando Fernandez, Jivko Sinapov, Raymond Mooney, and Peter Stone. Interaction and autonomy in robocup@home and building-wide intelligence. In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence and Human-Robot Interaction (AI-HRI)*, October 2018.
6. Yuqian Jiang, Nick Walker, Minkyu Kim, Nicolas Brissonneau, Daniel S Brown, Justin W Hart, Scott Niekum, Luis Sentis, and Peter Stone. Laair: A layered architecture for autonomous interactive robots. In *Proceedings of the AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)*, October 2018.
7. Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+p: Empowering large language models with optimal planning proficiency, 2023.
8. Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. BWIBots: A platform for bridging the gap between AI and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.
9. Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
10. Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.
11. David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

12. Woojin Chung, Hoyeon Kim, Yoonkyu Yoo, Chang-Bae Moon, and Jooyoung Park. The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder. *IEEE transactions on industrial electronics*, 59(8):3156–3166, 2012.
13. Noriyuki Kawarazaki, Lucas Tetsuya Kuwae, and Tadashi Yoshidome. Development of human following mobile robot system using laser range scanner. *Procedia Computer Science*, 76:455–460, 2015.
14. Matthias Scheutz, John McRaven, and Gy Cserey. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1347–1352. IEEE, 2004.
15. Nicola Bellotto and Huosheng Hu. Multisensor-based human detection and tracking for mobile service robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):167–181, 2009.
16. Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3908–3916, 2015.
17. Shengong Chen, Youfu Li, and Ngai M. Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 11(30):1343–1377, 2011.
18. Michele Colledanchise. *Behavior Trees in Robotics*. PhD thesis, KTH Royal Institute of Technology, 2017.
19. P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, March 2020. arXiv: 2003.09003.
20. Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.
21. Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for perceiving and processing reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019.

HSR Software and External Devices [DSPL]

We use a standard Human Support Robot (HSR) from *Toyota*. No modifications have been applied.

Fig. 3. HSR

Robot's Software Description

We are using the following 3rd party software:

- Object recognition: YOLO, SAM, and TensorRT
- People and activity recognition: OpenPose, MediaPipe, OSNET
- Manipulation: MoveIt
- Knowledge Base: PostgreSQL
- Planning and reasoning: Clingo, PDDLStream
- State Machine: SMACH (ROS)

External Devices

We are using the following external devices:

- Asus ROG Laptop (Backpack)
- MSI Laptop (Backpack)

Cloud Services

We are using the following cloud services:

- Speech recognition: Google Cloud Speech API
- Large language model: GPT4