SUTURO-VaB 2023 Team Description Paper

Vanessa Hassouna

November 27, 2023

Abstract. The team SUTURO VaB is a joint RoboCup@Home Domestic Standard Platform League (DSPL) team of the Artificial Intelligence (IAI) at the University of Bremen in Germany and the Technical University of Vienna (TUW). Our approach is based on joining the existing open source software stack for knowledge representation and reasoning, manipulation, perception and planning of the IAI with the novel approaches for exploration and perception developed by TUW.

1 Introduction

The SUTURO VaB team is a collaboration between the Institute of Artificial Intelligence (IAI) at the University of Bremen in Germany and the Vision for Robotics group at Technical University of Vienna (TUW) in Austria to address the challenges of everyday robotic activities in domestic scenarios. By joining forces, our team for RoboCup@Home combines state-of-the-art robotic perception (TUW) with state-of-the-art robot control architectures (IAI). TUW delivers a suite of perception modules that provide semantic information, such as object identities, locations and grasp configurations. The information is managed by IAI through a control architecture that builds a concise and accurate world representation that enables task planning based on semantic queries.

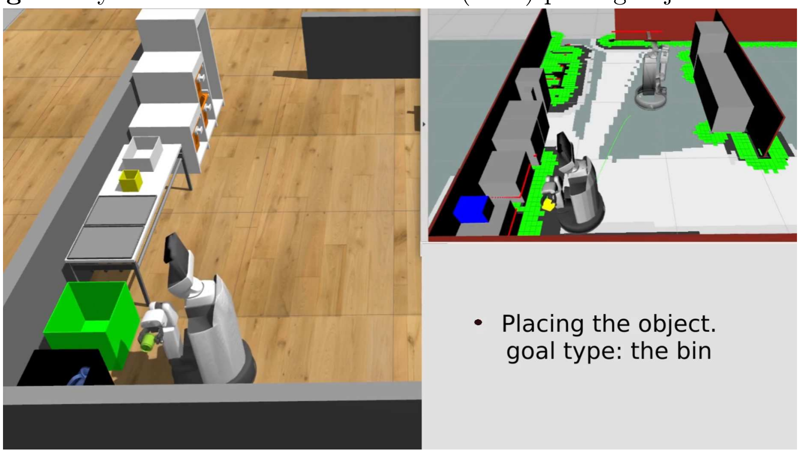
The collaboration between the IAI and TUW makes significant progress towards cognition-enabled service robotics. In particular, the integration of perception from TUW with the knowledge representation from IAI results in a highly sophisticated and capable robotic system. The IAI and TUW has a long-running relationship starting with the use of BLORT [1], developed at TUW, as a vision component in RoboKudo[2], developed at the IAI, all the way to collaboration on ongoing research projects such as Knowledge4Retail and TraceBot. The team entry for RoboCup@Home builds on the existing interface and endeavors to improve the overall system by combining state-of-the-art perception and the latest developments in unstructured information management and knowledge-based robotics made by IAI.

In general, our system has the capacity to be used for a wide variety of domestic robotic tasks in the open world. This is important not only for making groundbreaking research but also for developing systems for practical everyday use. For the purpose of the RoboCup@Home competition, the system is tuned for the "Store Groceries" and "Clean up" challenges in stage I and "Set the table" challenge in stage II. Furthermore we deeloped processes for the "Recepetionist" and "Carry my Luggage" tasks.

1.1 Previous experience in RoboCup@Home

SUTURO has participated in the domestic league of the RoboCup@Home German Open in 2019 for the first time. SUTURO included five Master students, supported by four PhD students. The code base of the team builds on top of several open source software projects developed and maintained at the IAI, including systems dedicated to knowledge representation and reasoning, motion planning, perception, and motion control. Due to time constraints, the SUTURO team only prepared for the Storing Groceries task where objects had to be transported from a table to a sensible location in a shelf. The team made it to the second stage, but did not proceed further. At the RoboCup@Home 2024 we got 7th placed and it was our first time being at the worlds. We are looking forward to increase our efforts towards the RoboCup.

Fig. 1. Toyota Human Assistance Robot (HSR) placing object in the bin



2 Approach to solving domestic tasks

This chapter will briefly describe the approach taken to solve domestic challenges including the technologies used in the process.

2.1 System description

Our system is based on our previous publication "The Robot Household Marathon Experiment" [3], which in turn is based on the one used in our previous RoboCup@Home participations. In that publication, we solve tasks very similar to "Clean up" and "Set the table", but in a known environment and on a PR2 robot. The architecture consists of four major components, for which we have implemented open source frameworks with tutorials:

- A plan executive (CRAM) [4],
- a motion planner (Giskard) [5],
- a perception framework (RoboKudo) [2],
- and a knowledge and reasoning engine (KnowRob) [6].

The plan executive is the core of the system. It defines high level goals for the robot and breaks them down into a sequence of motion, perception or reasoning tasks to achieve the overall goal. CRAM plans are written generically and are parameterized at runtime by utilizing various reasoning mechanisms within CRAM itself, or calling external perception, knowledge and manipulation services. These parameters describe e.g., what the navigation goal is, where an object should be placed or grasped from, or how the next motion should be constrained. Motions are described as a combinations of subgoals, that the motion planner Giskard offers, e.g., "keep the cup you are holding level, while also driving to a target pose and avoiding collisions". Using KnowRob's knowledge base, the plan executive exploits formal knowledge about the environment, the embodiment of the robot and the relationships between them. This has been originally designed for robots acting in domestic environments, and thus comes with rich information about this domain.

Interaction between CRAM and the perception system is realized through RoboKudo (see 3.3). It receives high-level perception tasks stated by CRAM and translates these into task-specific perception pipelines, that employ an ensemble of experts. These sequences of vision methods are mainly driven by individual experts called *Annotators*, which are connecting a specific vision method into the overall framework. The methods provided by TUW are therefore integrated as annotators into the RoboKudo perception framework allowing the reasoning about the perception results and the communication with the high-level planning CRAM to close the perception-action loop.

2.2 Application use case: Storing Groceries

In the following we will describe how our system is able to solve a common task for cognitive agents, namely to store groceries into their corresponding places.

Fig. 2. Toyota Human Assistance Robot (HSR) scanning the shelf

Task 1: Setup Our system will start by autonomously creating a semantic mapping of the environment based in the domain rich knowledge representation provided by KnowRob. To move the robot to the most promising, generated viewpoint, the pose is passed to Giskard as a goal pose.

Task 2: Scan table After reaching a viewpoint, the CRAM plan triggers the RoboKudo pipeline. The pipeline invokes the vision methods from TUW and IAI. Pipeline results contain the object information like color, confidence and class label, for every perceived object. The perceived objects are then validated against the knowledge base in KnowRob, which instantiates objects corresponding to their taxonomy. Lastly, the agent returns into a neutral pose to prepare for grasping.

Task 3: Grasping The Agent starts with the item that is located closest to it and moves into a position from where the item can be grasped. The knowledge base is queried by the CRAM plan for dimensions and position of the object, which is then used to set a motion goal using Giskard. Following this motion, we further check if the gripper is closed properly around the item, which in turn results into updating the agents beliefstate.

Task 4: Scan shelf The Agent will now move into a suitable position and pose in order to scan the shelf. After the movement is completed, the perception pipeline is called. The shelf is represented as multiple layers. Each layer will be processed individually in the perception pipeline. The perceived objects and their layer identification are then stored in the knowledge base.

Task 5: Place sequence The handled item must now be placed at a proper position on the surface. We infer the position yet again from our KnowRob knowledge base, where the corresponding taxonomy of the grasped object lets us infer the proper placement. The position is then passed over to Giskard in order to place the item. After successfully placing the targeted item, the robot will then continue to work on the remaining objects.

3 Innovative technology and scientific contribution

In this section we will describe our scientific contributions and open source frameworks in more detail, which we will be employing at the RoboCup@Home.

3.1 CRAM

CRAM (Cognitive Robot Abstract Machine) [4] is an open source toolbox for designing, implementing and deploying software on autonomous robots. The framework provides various tools and libraries for aiding in robot software development as well as geometric reasoning and fast simulation mechanisms to develop cognition-enabled control programs that achieve high levels of robot autonomy. CRAM also provides tools for introspection that enable the robots to reason about their past executions and improve by autonomously optimizing their plans.

3.2 KnowRob

KnowRob (Knowledge processing for Robots) [6] is an open source knowledge processing system designed for autonomous service robots that addresses several aspects that are commonly not sufficiently considered in AI KR&R systems. One of these aspects is that robots need a more fine-grained action representation. This is because service robots should be able to cope with (often) shallow and symbolic instructions, and to fill in the gaps to generate detailed, grounded, and (often) real-valued information needed for execution.

Recently, the focus has shifted towards integrating simulation and rendering techniques into a hybrid knowledge processing architecture. The rational is to reuse components of the control program in virtual environments with physics and almost photorealistic rendering, and to acquire experiential knowledge from these sources, which can be used to draw conclusions about what action parameterization is likely to succeed in the real world (e.g., through learning methods).

3.3 RoboKudo

RoboKudo is the second version of RoboSherlock [2], which is an open source robotic perception framework based on the principles of unstructured information management. The framework allows for the creation of perception systems that employ an ensemble of experts approach and treat perception as a question-answering problem. Based on the queries issued to the system a perception plan is created consisting of a list of experts to be executed. The perception experts generate object hypotheses, annotate these hypotheses and test and rank them in order to come up with the best possible interpretation of the data and generate the answer to the query. We are also currently investigating new and more flexible perception process models as well as a general framework advancement which will be driven by the RoboCup challenges.

Semantic mapping and exploration Based on our previous work [7], we propose to let the robot autonomously decide how to explore the room. After mapping the room, a fast and straightforward process, the robot infers the different furniture present in each room and extract relevant reachable viewpoints to investigate each area of interest of every relevant furniture. These viewpoints can then either be all explored using a travelling salesman solver, or visited based on the current knowledge for each area (likelihood of an object to be present there)

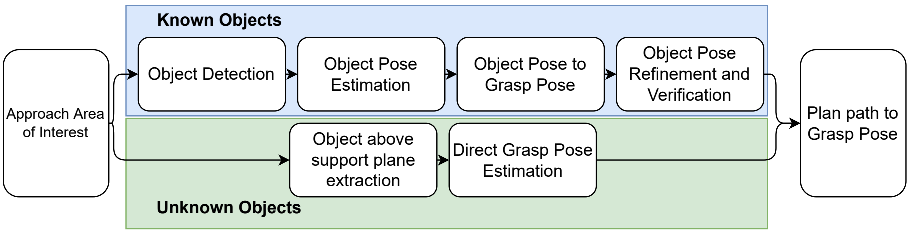


Fig. 3. Perception pipeline for object understanding.

Object understanding for manipulation Object understanding is essential for any autonomous manipulation. We propose to follow a pipeline based on object detection, object pose estimation and object pose refinement and verification to drive the manipulation, as illustrated in Figure 3. The method for each part was chosen based on its robustness and computational constraints. In particular, for object detection, Mask-RCNN [8] has proven to be very reliable when knowledge of objects are available. For pose estimation, Point Pair Features-based (PPF) methods are very robust and scalable when combined with powerful object detector like Mask-RCNN. They also require minimal training time and very reasonable computational resources, especially when exploiting symmetries of objects as presented in our previous work [9]. The only drawback comes from their limited accuracy, which can be compensated by the use of pose refinement as demonstrated in VeRefine [10]. VeRefine combines physical simulation and Iterative Closest Point (ICP) such that they prevent each other from diverging, all the while supervising the process by a rendering-based scoring that verify the validity of the final result. This lets us recover an accurate and physically plausible pose, from which manipulation can be performed with increased safety and confidence.

Unknown object manipulation The pipeline presented up until now uses some level of object knowledge. This is a natural requirement in the context of tidy-up applications where the robot has to deal with a slowly evolving set of objects, and can therefore build knowledge about them. There are however scenarios where neither the specific object nor the category of an object can be known. To deal with such situation, we propose to use HAF-grasping [11] as a direct grasp pose estimation method, given its robustness and limited computational requirements. This method can be used whether the category of the object is known or not, guaranteeing the ability of the robot to pick up objects in its way. This pipeline is illustrated in Figure 3.

3.4 Giskard

Giskard [5] is an open source motion planning framework. It uses constraint and optimization based task space control to generate trajectories for the whole body of mobile manipulators. Giskard offers interfaces to plan and execute motion goals and to modify its world model. A selection of predefined basic motion goals can be arbitrarily combined to describe a motion. In contrast to the most popular alternative MoveIt![12], Giskard will produce efficient trajectories deterministically, by employing a controller internally, instead of a sampling based motion planner. In addition, Giskard views the whole world as a single controllable system. This enables us to describe motions by constraining the environment, e.g., the opening of a door can be expressed as a joint goal for the door's hinge, combined with a Cartesian goal, to keep the robot's gripper at the handle.

4 Conclusions and future work

In this paper we presented the existing software frameworks and results of the IAI and TUW. We described how they will help us solve the challenges of the RoboCup@Home DSPL 2023. We hope that our participation will result in an extended, deeper and application-driven integration of the the state-of-the-art robotic perception by TUW and state-of-the-art robot control architectures by IAI to build a capable cognition-enabled robot architecture for housekeeper activities. The goal is to publish these integrations in our open-source software repositories to make them available to the public and therefore provide a powerful and generalized approach for manipulation activities to the robotics community.

References

1. T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze. BLORT - The blocks world robotic vision toolbox best practice in 3D perception and modeling for mobile manipulation. In *Proc. of IEEE ICRA, Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

2. Michael Beetz, Ferenc Bálint-Benczédi, Nico Blodow, Daniel Nyga, Thiemo Wiedemeyer, and Zoltán-Csaba Marton. Robosherlock: Unstructured information processing for robot perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1549–1556. Citeseer, 2015.
3. Gayane Kazhoyan, Simon Stelter, Franklin Kenghagho Kenfack, Sebastian Koralewski, and Michael Beetz. The robot household marathon experiment. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9382–9388. IEEE, 2021.
4. Michael Beetz, Lorenz Mösenlechner, and Moritz Tenorth. Cram—a cognitive robot abstract machine for everyday manipulation in human environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1012–1017. IEEE, 2010.
5. Simon Stelter, Georg Bartels, and Michael Beetz. An open-source motion planning framework for mobile manipulators using constraint-based task space control with linear mpc. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022. To be published by October 2022.
6. Michael Beetz, Daniel Beßler, Andrei Haidu, Mihai Pomarlan, Asil Kaan Bozcuoğlu, and Georg Bartels. Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2018.
7. Edith Langer, Timothy Patten, and Markus Vincze. Where does it belong? autonomous object mapping in open-world settings. *Frontiers in Robotics and AI*, 9, 2022.
8. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
9. Sergey V Alexandrov, Timothy Patten, and Markus Vincze. Leveraging symmetries to improve object detection and pose estimation from range data. In *International Conference on Computer Vision Systems*, pages 397–407. Springer, 2019.
10. Dominik Bauer, Timothy Patten, and Markus Vincze. Verefine: Integrating object pose verification with physics-guided iterative refinement. *IEEE Robotics and Automation Letters*, 5(3):4289–4296, 2020.
11. David Fischinger, Astrid Weiss, and Markus Vincze. Learning grasps with topographic features. *The International Journal of Robotics Research*, 34(9):1167–1194, 2015.
12. 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.
13. Zhou Fang, Georg Bartels, and Michael Beetz. Learning models for constraint-based motion parameterization from interactive physics-based simulation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4005–4012. IEEE, 2016.
14. Ferenc Balint-Benczedi. *Task adaptable, Pervasive Perception for Robots Performing Everyday Manipulation*. PhD thesis, University of Bremen, 2020.

Software and External Devices [DSPL Template]

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

Fig. 4. Toya @ Insititute for Artificial Intelligence

Robot's Software Description

For our robot, we use the following Open Source robot frameworks developed by the IAI:

- High-level planning: CRAM[1]
- Motion planning: Giskard[2]
- Perception: RoboKudo[3]
- Knowledge processing: KnowRob[4]

and the TU Wien:

- Perception for grasping [5]
- Object pose refinement and verification [6]
- Grasp Pose Estimation [7]
- Plane pop-out [8]

Additional RoboCup@Home specific software is available as Open Source on Github[9]. In addition, we rely on the following third party Open Source software:

- Platform: Ubuntu 20.04
- Middleware: ROS noetic
- ROS navigation stack
- move_base

External Devices

Our robot relies on the following external hardware:

- Lenovo Legion Pro 7-16IRX G8
  Intel® Core i9-13900HX 24-Core (bis 3.90/5.40 GHz),
  1TB PCIe 4.0x4 SSD, 32GB RAM, NVIDIA® GeForce® RTX 4080 mit 12GB

Cloud Services

Our robot does not connect to any cloud services.

[1] http://cram-system.org/
[2] https://github.com/SemRoCo/giskardpy
[3] https://robokudo.ai.uni-bremen.de/index.html
[4] http://knowrob.org/
[5] https://github.com/v4r-tuwien/grasping_pipeline/
[6] https://github.com/dornik/verefine
[7] https://github.com/v4r-tuwien/haf_grasping
[8] https://github.com/v4r-tuwien/table_plane_extractor
[9] https://github.com/SUTURO/ and https://github.com/v4r-tuwien/