Australian Centre for Robotic Vision RoboCup@Home Team Description Paper

Steven Martin[1], Gavin Suddrey[1], Ben Talbot[1], Rohan Smith[1],
Feras Dayoub[1], and Niko Sünderhauf[1]

[1]Queensland
University of Technology, 2 George Street, Brisbane, QLD Australia,
https://roboticvision.atlassian.net/wiki/spaces/Cup/overview

February 17, 2019

Abstract. This paper describes the service robot Rocky from Team ACRV for the 2019 RoboCup@Home Challenge. This platform aims to demonstrate current Centre research in the context of mobile manipulation and human interaction. The system plans to use a behaviour tree style agent for autonomous control and leverage our existing expertise from the Amazon Robotics Challenge and other research to build and demonstrate real-world mobile manipulation and human interaction abilities.

1 Introduction

Team Australian Centre for Robotic Vision (ACRV) was the winner of the Amazon Robotics Challenge in 2017 and aims to participate in the Sydney 2019 RoboCup@Home Challenge. The team is based at the Queensland University of Technology and is a collaborative effort from university nodes within the Centre. The team is made up of undergraduate students, PhD students, professional staff, and research staff. The team leverages the related research experience of its members, past participation in international competitions, and rapid prototyping abilities to enable iterative, innovative and low-cost solutions.

Our solution and platform uses a number of software packages developed as part of the Centre's research including our grasping network[1], indoor navigation improvements to ROS move-base used on Softbank's pepper[2], and experience from the Amazon Robotics Challenge[3]. All other hardware, including our current platform, and software architecture has been created after the expression of interest put forward on January 20 2019. This paper describes the overall system architecture, manipulation capabilities, mobile navigation and mapping systems, offboard image recognition software, open source software and other contributions.

2 Team ACRV@Home

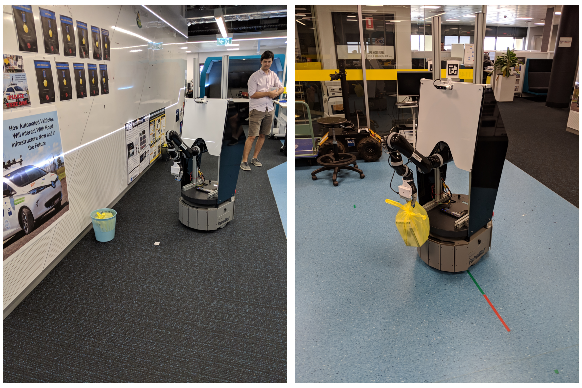

Fig. 1. Rocky carrying a bag while completing the Clean Up challenge task

With significant expertise in mobile navigation, SLAM, object detection, and manipulation, the team is participating in the RoboCup@Home competition to further their research in the domain of mobile manipulation. The team plans to use this experience to enhance the knowledge base of both incoming and continuing PhD students with support from engineers and senior researchers. We plan to release the competition software stack as open source software upon the completion of the competition and to attend future competitions in 2020 and beyond.

2.1 Focus of research

The Centre has significant manipulation experience from winning the 2017 Amazon Robotics Challenge and ongoing research output on new grasping techniques. These techniques however have been limited to applications in the tabletop manipulation space. The key goal of the team is to transition the existing research from the fixed position tabletop space to a real world mobile manipulation capability.

3 Overall System Design

The software architecture of the system, shown in Figure 2, splits the components into three overarching groups: the world model, robot actions, and cloud services. The robot agent sits at the top of the software stack, using a behaviour tree to guide the robot through the completion of complex tasks. The behaviour tree interacts with the world state and robot through service requests and robot actions. Robot actions also leverage a number of off board services, provided through CloudVis, to facilitate algorithms with computational requirements beyond what is possible to house on a mobile robot.

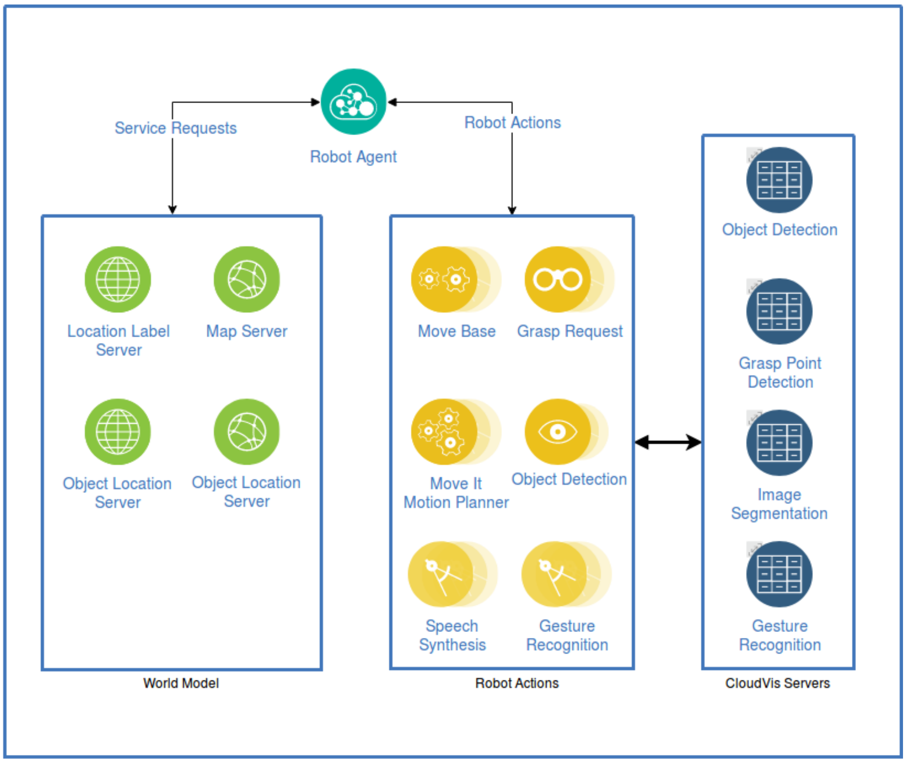

Fig. 2. Current software architecture design for Rocky

The current robotic hardware has been assembled from existing resources. Figure 1 shows our current platform, fondly known as Rocky. It has a flexible and modular design allowing for rapid changes to manipulator position, additional sensor placements, and structural reconfiguration. Due to the benefits of a flexible and modular design, there is expected to be continued iteration on the look and abilities of this platform during the months leading up the competition.

3.1 Hardware

Rocky is built upon a legacy Mobile Robotics PatrolBot base which has been modified to remove the onboard computer, add additional battery capacity, and mounting points for customisable structures. The internal structure of Rocky is shown in Figure 3. The robot is equipped with an onboard Mini-ITX desktop computer with an Intel i5 8600 @ 3.1GHz and 8GB of Ram, running Ubuntu 16.04 and ROS Kinetic.

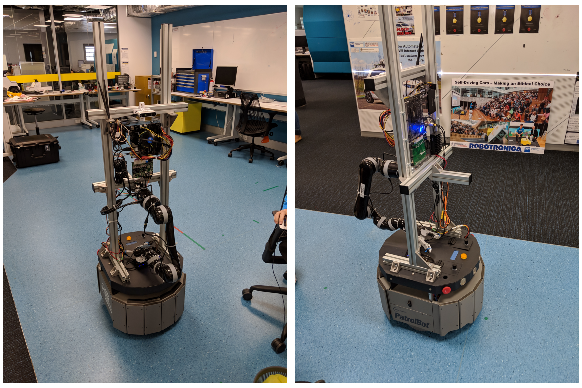

Fig. 3. Rocky without covers shows the construction of a 4040 extrusion which allows easy modifications of the structure, sensor placement, and arm orientation and location.

For manipulation, the robot is equipped with a 6-DOF robotic arm (Kinova Mico). The end effector is currently the stock Kinova fingers, however investigations into re-using design elements from the Amazon Robotics Challenge [4] are ongoing.

The current sensor suite consists of a SICK laser scanner for obstacle avoidance, and two Intel RealSense D435 cameras mounted forward facing and as a depth in hand camera.

3.2 Behaviour Tree Agent

Autonomous operation of our platform is achieved through the use of a custom built agent that utilises behaviour trees. Behaviour trees are a representation format for generating complex behaviours in autonomous agents, such as video game characters or robotic agents [5]. The decision to employ behaviour trees was driven by both their modular nature and their expressive power, which has been shown to be equivalent to that of finite-state machines. An example behaviour tree can be seen in Figure 4.

Formally, a behaviour tree is a rooted directed tree structure consisting of at least 1 or more nodes. These nodes can be classed into two groups: composite nodes – which contain one or more child nodes, and leaf nodes – which represent conditions and actions that are executable by the agent. At each time step, the agent sends a tick to the root node of the tree, which is then filtered through the tree based on the policy of each of the nodes within the tree. The tree then returns the current execution status of the tree to the agent, where the status is one of the following: success, which indicates the successful completion of the tree; failure, which indicates a non-recoverable failure at some point within the tree; and running, which is returned while the tree has yet to succeed or fail. How a node handles a tick, as well as how it hands return values back up the tree, is dependent on its type.

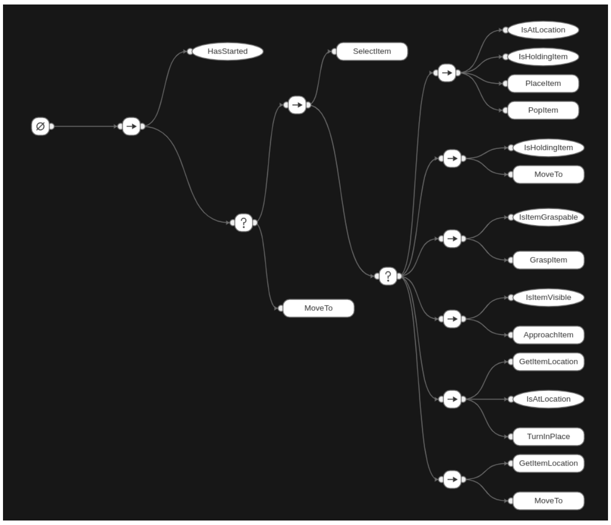

Fig. 4. Example behaviour tree model for Clean Up task.

To look at how a tick is handled by a composite node, we will look at each of the two basic forms of composite nodes: the Sequence node (represented as an arrow), and Selector node (represented as a question mark). When a sequence node receives a tick from its parent, it will respond by ticking each of its children in order until either it has exhausted all of its children (in which case it will return success) or a child returns failure or running (in which it will return either failure or running respectively). In contrast a selector node will tick each child successively until a child returns either success or running, at which case it will return success or running respectively, or it will return failure when it has exhausted its set of children. As a result of these dynamics, selector nodes can be used in a fashion similar to an if-else branch while sequences provide directional execution of actions.

Leaf nodes in a tree represent either conditions or actions executable by the agent. Unlike composite nodes, leaf nodes can have no children. Condition nodes return either success or failure depending on whether the particular condition is true or false, and should never return running. Action nodes, which represent actionable behaviours of the robot such as grasping or moving, return success if the action completes successfully, failure if the action failed, or running if the action is currently in progress.

In addition to composite nodes and leaf nodes, some behaviour tree formalisms also allow for additional nodes known as decorators. Decorators differ from composite nodes in that they may only contain a single child, but like composite nodes cannot be used as leaf nodes within a tree. Decorator nodes operate by modifying the return value of their children, and can be used for purposes such as inverting the success/failure return status of its child, or for more complex structures such as "for" loops.

3.3 Manipulation

Our grasping approach is based on Centre research for developing a Generative Grasping Convolutional Neural Network (GG-CNN). The GG-CNN is a lightweight, fully-convolutional network which predicts the quality and pose of antipodal grasps at every pixel in an input depth image. The lightweight and single-pass generative nature of GG-CNN allows for fast execution and closed-loop control, enabling accurate grasping in dynamic environments where objects are moved during the grasp attempt. Detailed explanation of this work is available in [1], and further extended to a multi view grasping approach in [3]. Software for training and a pretrained network implementation are available at GitHub[1] along with wrappers for use on the Kinova Mico[2].

3.4 Object Classification

Significant work was done in the development of a pipeline for rapid learning of new items for the Amazon Robotics Challenge. Details of the pipeline, approach and implementation can be found in [6]. We intend to use a modified version of this pipeline for detecting objects in competition.

3.5 Navigation and Mapping

Our navigation approach is based on the Standard ROS G-Mapping and obstacle avoidance implementations. We also incorporated some of the lessons and improvements used to enable Softbank Pepper to perform tours of our lab environment. Details of the modifications can be found in [2]. The QUT lab also has a long history of mobile navigation research, including: long-term navigation [7], using multiple map hypotheses for non-stationary environments [8], and applying vision to topometric maps for autonomous navigation [9].

Both the Australian Centre for Robotic Vision, and team members from within QUT, have independently developed research exploring how symbolic and semantic information can be used to enhance the navigation capabilities of mobile robots. From this research two approaches have emerged for the challenging problem of reconciling the informative utility of spatial symbols with the real world grounding provided by sensorimotor observations.

The first approach, Semantic Mapping and Semantic SLAM, looks at how a synergistic relationship can be constructed between the semantics in object detection and the metricity of traditional SLAM techniques. The benefits of combining object recognition with SLAM are bidirectional; the semantics of object labels provides a richer SLAM representation of the surrounding environment,

[1] https://github.com/dougsm/ggcnn

[2] https://github.com/dougsm/ggcnn_kinova_grasping

while the metric information provided by traditional SLAM enhances the spatial quality of object recognitions. The capabilities produced from this research – object-oriented SLAM [10], semantic mapping [11], and place categorisation [12] – will help the team to successfully use the metricity of the challenge environments to enhance object recognition capabilities, and the semantics in object recognition to produce rich spatial models when completing tasks.

The second approach to using symbols in navigation goes a step further, and looks at how a robot can use the symbolic spatial information communicated by human navigation cues to purposefully navigate previously unseen environments. Navigation cues can include labels, signs, maps, planners, spoken directions, and navigation gestures. The research uses a novel tool called the abstract map [13] to construct a symbolic representation of space, which can be tethered to the sensorimotor spatial model developed by a robot. The deformable nature of the links and tethers in the abstract map allows the robot to convert a navigation task defined solely with symbols into the real world navigation actions required to complete the task. The abstract map is available to help the team deal with tasks which may require the challenge of purposefully operating in unseen environments.

3.6 Speech Recognition

We are currently testing the PicoVoice[3] Porcupine and Rhino wake word and speech-to-text software available for non-commercial use. This provides a deep learning speech recognition approach with minimal hardware requirements that is capable of running onboard. Additional investigations into cloud based software are ongoing, but due to internet connectivity requirements and issues with connectivity an onboard or local network solution is preferred.

3.7 Offboard Image Processing

The Centre has developed software to allow low latency offboard image processing known as CloudVis. A demonstration of this software can be seen via its web interface[4]. This interface abstracts some of the complexities of the ROS publisher-subscriber structure; simplifies the use of algorithms through web, Python, MATLAB, C++, and other interfaces; and has high throughput with minimal overhead.

4 Conclusion

In this paper we have given an overview of the software and hardware to be used by Team ACRV at the RoboCup@Home challenge. We have presented our rapidly constructed robot, custom software packages, and overall architecture

[3] https://github.com/Picovoice

[4] http://cloudvis.roboticvision.org

that we will employ in tackling the challenge. We hope to build on our successes in previous competitions, and mature the mobile manipulation research within the Australian Centre of Robotic Vision. In the minimal time that the team has been preparing we have made significant progress towards what we believe is a competitive system.

References

1. Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *CoRR*, abs/1804.05172, 2018.
2. Gavin Suddrey, Adam Jacobson, and Belinda Ward. Enabling a pepper robot to provide automated and interactive tours of a robotics laboratory. *arXiv preprint arXiv:1804.03288*, 2018.
3. Douglas Morrison, Peter Corke, and Jürgen Leitner. Multi-view picking: Next-best-view reaching for improved grasping in clutter. *arXiv preprint arXiv:1809.08564*, 2018.
4. S Wade-McCue, N Kelly-Boxall, M McTaggart, Douglas Morrison, Adam W Tow, J Erskine, R Grinover, A Gurman, T Hunn, D Lee, et al. Design of a multi-modal end-effector and grasping system: How integrated design helped win the amazon robotics challenge. *arXiv preprint arXiv:1710.01439*, 2017.
5. Michele Colledanchise and Petter Ögren. Behavior trees in robotics and AI: an introduction. *CoRR*, abs/1709.00084, 2017.
6. Anton Milan, Trung Pham, K Vijay, Douglas Morrison, Adam W Tow, L Liu, J Erskine, R Grinover, A Gurman, T Hunn, et al. Semantic segmentation from limited training data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1908–1915. IEEE, 2018.
7. Feras Dayoub, Timothy Morris, Ben Upcroft, and Peter Corke. One robot, eight hours, and twenty four thousand people. In *Proceedings of the 2013 Australasian Conference on Robotics and Automation*, pages 1–7. Australasian Robotics and Automation Association (ARAA), 2013.
8. Timothy Morris, Feras Dayoub, Peter Corke, Gordon Wyeth, and Ben Upcroft. Multiple map hypotheses for planning and navigating in non-stationary environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2765–2770. IEEE, 2014.
9. Feras Dayoub, Timothy Morris, Ben Upcroft, and Peter Corke. Vision-only autonomous navigation using topometric maps. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1923–1929. IEEE, 2013.
10. L. Nicholson, M. Milford, and N. Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, Jan 2019.
11. N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085, Sep. 2017.
12. N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford. Place categorization and semantic mapping on a mobile robot. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5729–5736, May 2016.

13. Ben Talbot, Obadiah Lam, Ruth Schulz, Feras Dayoub, Ben Upcroft, and Gordon Wyeth. Find my office: Navigating real space from semantic descriptions. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5782–5787. IEEE, 2016.
14. Douglas Morrison, Peter Corke, and Juergen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: Science and Systems 2018*, Pittsburgh, Penn, 2018.

Name of team: Team Australian Centre of Robotic Vision
Members: Steve Martin, Ben Talbot, Gavin Suddrey, Rohan Smith, Feras Dayoub, Niko Sünderhauf
Contact Information: steven.martin@qut.edu.au
Website: `https://roboticvision.atlassian.net/wiki/spaces/Cup/overview`

Hardware:

- MobileRobot PatrolBot base
- Kinova Mico
- Onboard computer Intel i5 8600, 8gb Ram, Ubuntu 16.04
- Sick Laser Scanner
- 2 x Intel Realsense D435

External Hardware:

- 3 x Desktop Workstations

Software:

- ROS
- OpenCV
- PCL
- MoveIt
- Tensorflow
- Picovoice Porcupine
- Picovoice Rhino
- Matlab