UNSW RoboCup@Home SPL
Team Description Paper

C. Sammut, M. Pagnucco, G. Castro, C. Flannagan, M. Gratton, B. Hengst, D. Rajaratnam, C. Schwering, M. Thielscher, M. Velonaki, T. Wiley, O. Natan, and V. Wong

School of Computer Science and Engineering & Creative Robotics Lab,
The University of New South Wales, Sydney, Australia,
`https://www.engineering.unsw.edu.au/computer-science-engineering/research/research-activities/robocup-at-home`

1 Introduction

The UNSW team took delivery of its HSR in April 2017 and took part in RoboCup 2017, although with limited success because of the short preparation time. Since then, we have become much more familiar with the platform, developing the skills necessary for the @Home competition and using the HSR in graduate classes.

This team description paper is accompanied by a video demonstrating some of our capabilities on a makeshift platform, put together specifically for this HSR in our newly setup up HRI laboratory, which is fitted out as a studio apartment.The demonstration includes:

- Mapping and Navigation
- Response to speech commands
- Recognising and grasping objects on a table
- Sound Localisation
- Person tracking and following

The University of New South Wales (UNSW) has a long history in RoboCup soccer and rescue leagues. Our main research focus in all of our participation in RoboCup has been on the AI underpinning intelligent behaviours. RoboCup@Home SPL with the Toyota HSR robot fits very well with our research focus, as the @Home competition demands more high-level reasoning and learning than any other league. The research conducted in the School of Computer Science and Engineering and the Creative Robotics Lab spans many areas including: cognitive architectures, machine learning for perception and robot behaviours, human-robot interaction (including conversational and multi-modal interaction), SLAM, and cognitive robotics. The diversity of our research gives us a good understanding of how to build a complex robot and we are experienced in integrating systems ready for competition, and in releasing our code as open source software. We also have unique expertise in the Creative Robotics Lab, which is dedicated to research in human-robot interaction and social robots.

The general theme of our work is on human-robot interaction and trust in the robot. Another agent is trusted if its behaviour is predictable. That is, each agent must build a model of the other agent that is accurate and reliable. Part of our work is in this model building. Another related study is in multimodal human-robot interaction. In particular, we can use the robot's SLAM system, and episodic memory to give the robot spatio-temporal awareness. This knowledge can be used to assist in language understanding. For example, if the language alone is not sufficient to disambiguate between a reference to an object, proximity, function or regency can be used as reasonable guesses to resolve the reference. To achieve this, the robot requires mapping at several levels of abstraction. The lowest level is the occupancy grid created by SLAM. On top of that, we require a topological map to associate spaces to names and relations. These can then be turned into logical predicates and reasoning applied within a logic framework. Connecting spatial reasoning to language understanding is the topic of a current postgraduate research project.

2 Background

We have a substantial code base inherited from the RoboCupRescue Real Robots competition and other research. The software is built around ROS and has been ported to run on a variety of platforms including robots with different drive mechanisms, sensors and arms. The existing software includes SLAM and autonomous navigation; multi-modal interaction for conversational agents; and software for object recognition and simple grasping. We also incorporate our current research in cognitive hierarchies and resource constrained planning and reasoning. The remaining components, such as inverse kinematics for manipulation, and face recognition are derived from existing open source software, especially pre-built ROS packages.

2.1 SLAM and Navigation

Our GPU accelerated 3D SLAM software for mapping and navigation [1] has been ported to the HSR. The SLAM system was developed to handle the complex terrain of urban search and rescue, such as going up and down stairs and navigating over uneven flooring. It avoids temporary obstacles, such as human occupants moving around. Some adaptation is required to deal with furniture, glass and mirrors. The navigation system includes exploration for mapping, as well as path planning [2].

An issue that we had in 2017 was that the IMU did not function on delivery. This, along with errors in the ROS software shipped with the HSR prevented the use of the 2D/3D GPU accelerated SLAM software developed as a PhD project at UNSW. When the IMU was repaired at RoboCup, there appeared to be unexpected interactions between the HSR ROS nodes and our SLAM code that caused SLAM to fail. At present, we are using the Google mapper, combined with our own exploration algorithm. We are still investigating why the GPU accelerated GraphSLAM is having problems.

2.2 Conversational Agent

A conversational agent was originally developed as part of a project to create a “smart home” [3]. The occupants interacted with devices in the home by speech and gestures. The system was also equipped with cameras to track motion, which was used to detect falls. Occupants were able to talk to the room and ask for devices to be turned on and off and to control television sets, audio systems, ask questions answered from the web, etc. The system consists of a scripting language, called FrameScript, for the dialogue and interacts with devices through a blackboard system. Each device is controlled by its own software agent that interacts with other agents, including the dialogue manager, through a blackboard. This system has been ported to the HSR, adding planning agents and other components needed for robot control. Agents interact with ROS nodes through the blackboard mechanism.

FrameScript can take its speech input from any speech-to-text system. We have tested it with PocketSphynx and, currently, we are using the Google speech API.

2.3 Robot Control and Reasoning

Our team includes experts in knowledge representation and reasoning (KRR), action logics, teleo-reactive programming, epistemic reasoning, and belief revision. This research is relevant, not only because of the planning required for the robot, but also because it must also be able to cope with incomplete or inaccurate statements from humans. For example, the human may ask for the red cup on the table when, in fact, there is a red plate and a blue cup. What should it do? We have worked to bring the theory of KRR to practice, helping develop ROSoClingo [4], an adaptation of a high-performance Answer Set Programming reasoner for use in ROS. We are implementing high-level reasoning and task planning in ROSoClingo. We are also experimenting with our own implementation of Nilsson’s Teleo-Reactive programming.

2.4 Object Recognition

For object recognition we have two approaches, one ”off-the-shelf” and the second which we are developing ourselves. The of-the-shelf method uses YOLO [5] to detect objects in the scene, placing bounding boxes around them and the using the point cloud from the RGB-D camera to locate the object in space. When attempting to grasp the object, we used ROS packages for finding the grasp points and planning the arm movement.

We have also developed model-based approaches to 3D object recognition using RGB-D cameras. The vision system extracts shape primitives (e.g. planes and cylinders) from the point cloud. A relational learning system then builds a description of the object class based on the relationships between the shape primitive [6]. This method has been used in the rescue environment to recognise staircases and other terrain features. Once a model of the object is created, it is imported into a simulator, like Gazebo, which allows the robot to "visualise actions" before executing them in the real world. We are also investigating other applications of 'logical vision" [7].

2.5 Externally available components

As indicated above, some components are derived from existing open source software, especially ROS packages. We use the MoveIt or Agile Grasp ROS packages for calculating inverse kinematics and performing manipulation tasks. For face recognition and person tracking we use tools in OpenCV 3.0, and the OpenNI/NiTE skeleton tracking library.

3 Research

One of our main research foci is on combining high-level reasoning with real-time low-level sensing and control to improve the capabilities of autonomous robots. Our long-term aim is to develop general-purpose intelligent systems that can learn and be taught to perform many different tasks by interacting with their environment. In the course of our research, we have created software that can be ported to the Toyota HSR for the RoboCup@Home competition. Below, we highlight the current focus of our research, and our key innovative technologies and scientific contributions.

3.1 Cognitive Architecture

We wish to better understand how a variety of software components should be integrated in a robot. We have developed a novel meta-model for formalising cognitive hierarchies [8]. A cognitive hierarchy consists of a set of nodes connected in a hierarchical graph. Every node in the hierarchy has a world model and behaviour generation at a particular level of abstraction, with the lowest-level node as a proxy for the external world. Cognitive hierarchies described using this model are modular in design and allow the integration of symbolic and sub-symbolic representations in a common framework. The model has been demonstrated on several platforms including a Baxter robot, which incorporates a simulator as its world model, allowing the system to "visualise" the effects of actions before executing them in the real world. For the TMC HSR RoboCup@Home SPL robot we will use this system, implemented over ROS, as the basis for integrating the different components in a single architecture, from SLAM and robot navigation through to high-level behaviour generation.

3.2 Human-Robot Interaction and Trust

Human-robot interaction may include speech, sound, music, gestures, body movements, proximity, facial expressions, body language and touch. Poorly designed interactions decrease the willingness of a human to use the robot. Our research aims to improve human-robot interaction by studying two areas, physical elements of human-robot interactions and the ability of the robot to learn from and adapt to new dynamics of the interaction.

The physical components of human-robot interactions we study are touch, gesture, and recognising human emotions through micro and macro human expressions, and the manner in which a robot approaches a human. [9] The goal is to prevent the human from being surprised or fearful of a robot's actions. We use machine learning to alter how the robot behaves and interacts so that the human can teach the robot how they wish to interact, explaining aspects of the interaction they prefer or dislike, find uncomfortable or confronting.

An associated concern is how trustworthy humans regard a robot, especially when they can learn and adapt to new situations. We are studying the change in trust for a mixed initiative task under varying degrees of transparency of the adaptation process. The cognitive architecture mentioned above includes the ability for the robot to adapt to a change. It is implemented on a Baxter robot for a mixed initiative problem solving task where the environment changes, requiring the robot to adapt on the job. This also requires modelling and evaluating the evolving human-robot trust relationship as the robot learns.

For our research in Human-Robot Interaction we have constructed the National Facility for Human-Robot Interaction Research, which saw it's first use during "The Big Anxiety Festival" from October to November 2017. It will be a state-of-the-art facility for non-intrusive real-time measurement of the properties that are linked to human affect and intent.

3.3 Position Tracking and SLAM

We developed our own robust position tracking and SLAM algorithms [2], originally for RoboCupRescue, but which are also used on robots in our office space. A recently completed PhD student improved and re-implemented these algorithms to make use of a GPU using full 3D information to produce correctly aligned and accurate 3D maps [1]. Much of this work carries across to RoboCup@Home, since accurate 3D position tracking and mapping for navigation and obstacle avoidance through the home. Combined with our work on spatial reasoning, this also assists in planning and model-based object recognition.

3.4 Robot Learning

UNSW was known for its work in machine learning well before we began working in robotics. In fact, one of the main motivations for entering robotics is that it is such a rich source of data and problems that can be solved by learning. We have developed methods for learning how to traverse difficult terrain by learning from demonstration and through trial-and-error [10]. We combine learning abstract qualitative models with reinforcement learning, where the abstract layer constrains search in the lower-control layer to greatly, reduce the number of trials required. As mentioned earlier, we also make extensive use of machine learning in perception.

3.5 General Game-Playing Robots

Our group also has a history of success in General Game-Playing (GGP) competitions, and this expertise extends to robotics. Many domestic robot tasks have game-like properties, requiring the robot to reason about the goals of other agents as well as adapting to unexpected changes in the environment [11]. For example, a domestic robot tasked with fetching an item has to consider the possibility that the item may not be where it expects, or that the human operator may change locations after issuing the request. Viewing such a task as a game can provide a framework for improving robot behaviours.

4 Experiments and Results

In lieu of results obtained using the Toyota HSR, we briefly list some experiments conducted on other platforms

4.1 Human-Robot Interaction

Several conversation agent systems have been developed to interact with smart homes and robots. The system shown in Figure 1 is a speech operated robot arm that can be instructed to pick up objects of different colours and shapes. Each device is controlled by its own software agent, which posts messages to and reads from a blackboard.

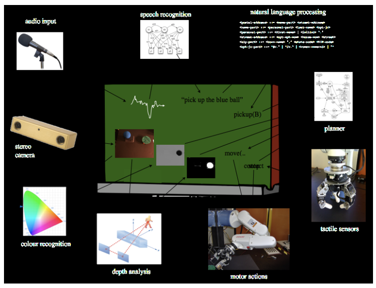

Fig. 1. Backboard for conversational agent

The same conversational agent architecture has been used to control a smart home (demonstrated in the accompanying video). Here, the agents attached to the blackboard control devices such as lights, the TV set, a radio and the home PC. Sensors include cameras and microphones monitoring a space. The system is capable of multi-modal interaction, combining gesture recognition with speech and can also perform safety monitoring, e.g. fall detection.

The conversational agent has been deployed in Sydney's Powerhouse Museum as a guide to its display on computing technology and its history. This installation was a valuable lesson in developing robust systems for the public. We learned that as long as visitor are cooperative and interested in learning about the museum, the system works well. However, we did not anticipate that the majority of visitors to the museum are school children whose main intent is to break the system! Thus, the system must be able to recover from unexpected interactions. Following that experience, the later software (FrameScript) provides mechanism for building recovery modes into the interaction.

4.2 Position Tracking and SLAM

Crosbot is the name of the SLAM system that has been under development for many years for the rescue robot competition. The UNSW team received the “best-in-class” award for autonomy three time, largely due to the accuracy of the maps. Most recently, these algorithms have been redeveloped to run on GPUs to speed up execution and to relieve the CPU of this work, enabling it to be used for other computations.

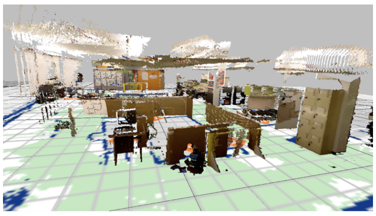

Fig. 2. 3D Map

The original 2D SLAM was extended to create 3D maps, fusing information from LIDAR and RGB-D cameras, as shown in Figure 2.

4.3 Robot Learning

Much of the research conducted by the UNSW team is focussed on robot learning. As described above, there has been a significant amount of work done on learning how to traverse irregular terrain, including climbing stairs [10].

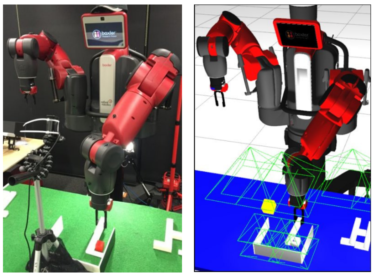

Fig. 3. Learning to use tools

Another current project gives the robot the ability to learn how to use objects as tools [12]. This uses symbolic machine learning methods to build theories of how objects of different shapes interact with other objects and reasoning about how to position and move them so that the object selected as a tool can allow the robot to complete a task that that it could not otherwise do, .e.g. using an object as a hook to pull another object out of a narrow space. The perceptual system builds models that are imported into a physics simulator, which is used to “visualise” actions before they are executed, thus extending the robot’s planning capabilities.

5 Conclusion and Future Work

As our RoboCupSoccer SPL teams have done over many years, we will make our software available to other research groups and teams competing in @Home. We are in the process of making our recently developed GPU-accelerated 3D SLAM publicly available as a ROS package. In addition, as a part of our involvement in @Home, we intend to publicly releasing the software (CrosBot) that we have developed in ROS for RoboCupRescue and which will be extended for use in RoboCup@Home.

All our research and development uses real robots. We have recently begun a collaboration with Fuji Xerox in Japan to investigate how intelligent social robots could be introduced into the workplace. The investigation will focus on the benefits that social robots could provide to employees, such as improving office well-being and productivity. Aspects of the investigation include studying human-robot interaction as the robot must understand and respond to requests in a manner that is comfortable for each user, incorporating real-time learning capabilities in the robot so that workers can teach the robot how they wish to interact with them or the ability for workers to teach the robot how to perform new tasks.

References

1. A. Ratter and C. Sammut. Local Map Based Graph SLAM with Hierarchical Loop Closure and Optimisation. In *2015 Aust. Conf. on Robotics and Automation*, 2015.
2. A. Milstein, M. McGill, T. Wiley, R. Salleh, and C. Sammut. A Method for Fast Encoder-Free Mapping in Unstructured Environments. *Journal of Fields Robotics, Special Issue on Safety, Security, and Rescue Robotics*, 28(6):817–831, 2011.
3. M. W. Kadous and C. Sammut. InCA: A Mobile Conversational Agent. In *8th Pacific Rim Int. Conf. on Artificial Intelligence*, pages 644–653, 2004.
4. B. Andres, D. Rajaratnam, O. Sabuncu, and T. Schaub. Integrating ASP into ROS for Reasoning in Robots. In *13th Int. Conf. on Logic Programming and Nonmonotonic Reasoning*, Lexington, USA, 2015.
5. Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *CoRR*, abs/1709.05943, 2017.
6. R. Farid and C. Sammut. Plane-based object categorisation using relational learning. *Machine Learning*, 94(1):3–23, 2013.
7. Stephen Muggleton, Wang-Zhou Dai, Claude Sammut, Alireza Tamaddoni-Nezhad, Jing Wen, and Zhi-Hua Zhou. Meta-interpretive learning from noisy images. *Machine Learning*, in press, 2018.
8. K. Clark, B. Hengst, M. Pagnucco, D. Rajaratnam, P. Robinsion, C. Sammut, and M. Thielscher. A Framework for Integrating Symbolic and Sub-symbolic Representations. In *25th Int. Joint Conf. on Artificial Intelligence*, New York, USA, 2016.
9. D. Silvera-Tawil, M. Velonaki, and D. Rye. Human-Robot Interaction with Humanoid Diamandini Using an Open Experimentation Method. In *24th IEEE Int. Symp. on Robot and Human Interactive Communication*, pages 425–430, 2015.
10. T. Wiley, C. Sammut, B. Hengst, and I. Bratko. A Planning and Learning Hierarchy using Qualitative Reasoning for the On-Line Acquisition of Robotic Behaviors. *Advances in Cognitive Systems*, 4:93–112, 2016.
11. D. Rajaratnam and M. Thielscher. Execution Monitoring as Meta-Games for General Game-Playing Robots. In *24th Int. Joint Conf. on Artificial Intelligence*, Buenos Aires, Argentina, 2015.
12. H. Wicaksono and C. Sammut. Relational tool use learning by a robot in a real and simulated world. In *Australasian Conference on Robotics and Automation*, 2016.

Annex

The foreground software used in 2017 has been made available on the Community Github at https://git.hsr.io/unsw/robocup2017. The software is described in the

README.md file of the git repository. An excerpt of the readme is provided below.

The ROS packages of the foreground software are described below.

- hsrb_unsw_behaviour - manages, at at task level, the current activity that the robot is executing.
- hsrb_unsw_database - tracks the internal memory of the robot, including the current map, and location of rooms and objects within the map. The database is integrated with other modules, such as the Clingo task planner.
- hsrb_unsw_framescript - contains the Framescript conversation files used in RoboCup@Home DSPL.
- hsrb_unsw_grasping - control and operation of the HSR arm for picking up objects.
- hsrb_unsw_launch - common launch files
- hsrb_unsw_manipulation - control and operation of the HSR arm.
- hsrb_unsw_robot_screen - outputs internal status messages to display on the HSR screen.
- hsrb_unsw_rqt - RQT plugins.
- hsrb_unsw_rviz - RViz Plugin for control of the HSR through RViz.
- hsrb_unsw_speech - PocketSphinx model files used in RoboCup@Home.
- hsrb_unsw_vision - Object recognigition and training files for use in RoboCup@Home.
- hsrb_unsw_vision_msgs - ROS messages for vision communication topics.

The conversational agent also uses the Google speech API and object detection incorporates YOLO.

UNSW has developed separately to use with the HSR and RoboCup@Home DSPL:

- CrosBot is a collection of ROS packages developed at UNSW for autonomous robots. This software includes:
  - 2D & 3D laser-based and encoder free position tracking for both CPUs and GPUs.
  - 2D & 3D laser-based and encoder free SLAM CPUs and GPUs.
  - Autonomous navigation and exploration for differential drive robots in unstructued terrains.
  - Common data structures for communication
- Framescript is a conversational agent developed at UNSW initially for use with a Smart Home, and then extended for use with autonomous robots. In accordance with the terms of the TMC Research Agreement, the Framescript conversation files used for RoboCup@Home have been released.
- UNSW has developed a filtered skeleton tracking algorithm, that has been integrated with OpenNI2 and NITE.
- ROS-o-clingo is a python interface between ROS and the Clingo4 ASP solver.