

KameRider OPL @Home 2017 Team Description Paper

Jeffrey Too Chuan Tan¹, Feng Duan², Zool Hilmi bin Ismail³ and Yutaka Uchimura⁴

¹The University of Tokyo, Japan

²Nankai University, China

³Universiti Teknologi Malaysia, Malaysia

⁴Shibaura Institute of Technology, Japan

jeffrey@iis.u-tokyo.ac.jp

<http://openbotics.org/kamerider/>

Abstract. This document is the team description paper of the KameRider OPL team for the participation of @Home Open Platform League in RoboCup 2017 Nagoya, Japan. KameRider team is a collaborative effort that aims to develop an open robot platform for service robotics. This paper describes the motivation of this effort, the hardware and software of the robot developments, and the scientific contribution and social impacts of our work via a new educational initiative – RoboCup @Home Education.

1 Introduction

KameRider team is a collaborative effort that aims to develop an open source robot platform for service robotics. Started from 2013, the limited development resources and manpower team condition had urged a strong motivation to develop a more affordable yet functional solution to take part in RoboCup @Home league and service robot development.

The current team objectives are as follows:

- A. Utilize open source solutions for both hardware and software to develop an open source robot platform that is affordable (low cost) and with large community support.
- B. Participate in RoboCup @Home to benchmark the open source robot platform performance.
- C. Support a new educational initiative – RoboCup@Home Education.

2 Background and Motivation

2.1 The challenges of RoboCup @Home

Starting from 2006, RoboCup @Home [1] has been the largest international annual competition for autonomous service robots as part of the RoboCup initiative. The challenge consists of a set of benchmark tests to evaluate the robots' abilities and performance in a realistic non-standardized home environment setting [2]. It has greatly

fostered artificial intelligence development in various domains including human-robot interaction, navigation and mapping in dynamic environments, computer vision, object recognition and manipulation, and many more developments on robot intelligence.

However, it is observed that the development curve of the RoboCup @Home teams have a very steep start. The amount of technical knowledge and resources (both manpower and cost) required to start a new team has made the event exclusive to only established research organizations. For instance, in domestic RoboCup Japan Open challenge, the participating teams in RoboCup @Home were less than 10 teams and similar teams ever since the past few years. There were actually several new team requests but the development gap was too huge for them to even complete the robots.

For this reason, our team had initiated the development of an open source robot platform for RoboCup @Home in 2013. The goal of the project is to develop a basic robot platform to facilitate startup team for the participation in RoboCup @Home. It is developed based on open source solutions for both hardware and software developments for low cost and large community support to facilitate startup of the novice teams. The first working prototype (Fig. 1) had participated in RoboCup Japan Open 2014 and had the honor to receive the *Japanese Society for Artificial Intelligence (JSAI) Award*. Along with the development, we are honored to obtain support from RoboCup Japan Committee and RoboCup Federation, and bonded collaboration among The University of Tokyo (Japan), Nankai University (China), Universiti Teknologi Malaysia (Malaysia) and Shibaura Institute of Technology (Japan).

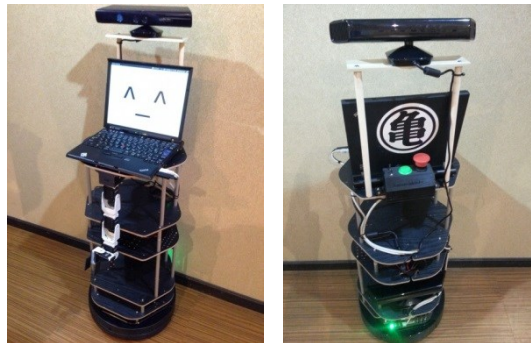


Fig. 1. The first working prototype of the open robot platform for RoboCup @Home

3 Robot Developments

3.1 Open source robot platform development

The open robot platform has a current basic robot hardware configuration (Fig. 2) for fundamental robot platform and add-on modular component systems for customized applications. For example in Fig. 3, a manipulator system (with top vision) and

an extended top vision system are added to the hardware configurations during RoboCup Japan Open 2015 for the applications in *Restaurant* task and *Follow Me* task.

3.2 TurtleBot as the basic robot hardware platform

TurtleBot¹ is a low cost (basic kit is approximately USD 1,000), personal robot kit with close integration to popular open source software, ROS² (Robot Operating System) [3]. The open source robot kit is adapted as the basic mobile platform for this development. The vertical range of the mobile manipulation can be adjusted with an elevated arm with linear motor, a secondary vision system is paired with the robotic arm for object recognition in the manipulation tasks, and 3D printed parts for component systems. An interactive interface with speech and facial expressions is in development for human-robot interaction. A general laptop PC (currently working on a single board computer system) with speakers and microphone is served as the main robot controller. Fig. 2 illustrates a basic robot hardware configuration for this development, and Fig. 3 shows add-on modular component systems, e.g. a manipulator system (with top vision) and an extended top vision system are added to the hardware configurations during RoboCup Japan Open 2015 and RoboCup 2015 Hefei.

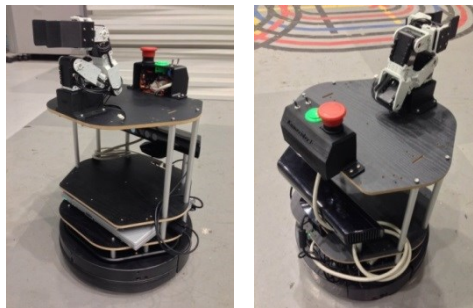


Fig. 2. The current basic robot hardware configuration



Fig. 3. Two hardware configurations during RoboCup Japan Open 2015 (left) and RoboCup 2015 Hefei (right)

¹ <http://www.turtlebot.com/>

² <http://www.ros.org/>

3.3 ROS as the robot software framework

ROS (Robot Operating System) is an open source robot software framework with a large community to provide huge collection of robotic tools and libraries. With ROS as the fundamental software framework, this work will adapt and assemble ROS packages and stacks to realize the navigation, manipulation, vision and speech functions of the robot in order to perform the tasks in RoboCup @Home.

Navigation. With the Kobuki³ and MS Kinect sensor as the mobile base hardware configuration, the TurtleBot navigation package⁴ is used for robot navigation with map building using *gmapping* and localization with *amcl*, while running the navigation stack in ROS. With the prebuild map and predefined waypoint locations, we can then instruct the robot to travel to a specific goal location with path planning using *actionlib*⁵.

Manipulation. We are using TurtleBot Arm⁶ for object manipulation (Fig. 4). It consists of 5 Dynamixel AX-12A servo motors, controlled by an *ArbotiX-M controller board/USB2Dynamixel*. While effort to integrate with *MoveIt!* is still in progress, we have integrated the arm control with object detection by color detection⁷ and object recognition by image processing for object manipulation. Once we recognized the object, we perform object localization by 3D point cloud to obtain the position of the object and calculate the inverse kinematic to make the movement of the arm to grasp the object.

Elevated Arm. An elevated arm (Fig. 5) is developed for flexible height manipulation. The current design is target to enable object manipulation at the height ranges from 0.3m to 1.8m.

Vision. A second vision system (Fig. 6) is built on top of robot with MS Kinect for people/object detection and recognition. The people tracking package is used to track people in the *Follow Me* task.



Fig. 4. Robot arm for object manipulation



Fig. 5. Elevated arm for flexible height manipulation

³ <http://kobuki.yujinrobot.com/home-en/>

⁴ http://wiki.ros.org/turtlebot_navigation/

⁵ <http://wiki.ros.org/navigation/Tutorials/SendingSimpleGoals>

⁶ http://wiki.ros.org/turtlebot_arm

⁷ <http://wiki.ros.org/cmvision>



Fig. 6. Vision system on top of the robot

Speech. For human voice interaction, we use CMU Pocket Sphinx⁸ as our robot speech recognizer (Fig. 7). It is a lightweight speech recognizer with a support library called Sphinxbase. We build our application with the latest version "sphinxbase-5prealpha". We use *gststreamer* to automatically split the incoming audio into utterances to be recognized, and offers services to start and stop recognition. The recognizer requires a language model and dictionary file, which can be automatically built from a corpus of sentences. For text-to-speech (TTS), we are using the CMU Festival system together with the ROS *sound_play* package.

In order to improve the speech recognition efficiency, we use a strategy for the robot to listen for activation keyword first, and once the keyword is recognized, it switches to *ngram* search to recognize the actual command. Once the command has been recognized, the robot can switch to grammar search to recognize the confirmation, and then switch back to keyword listening mode to wait for another command.

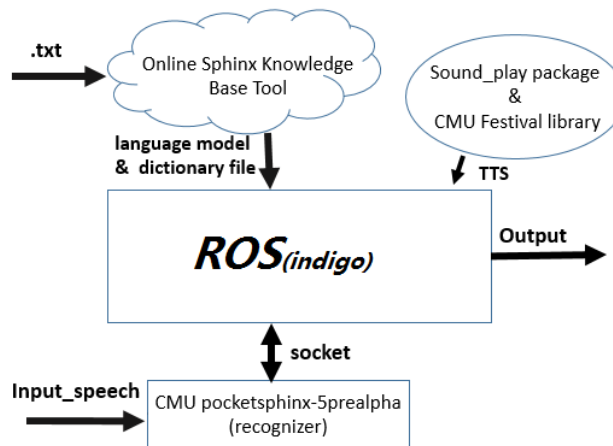


Fig. 7. The speech recognition and synthesis system

⁸ <http://wiki.ros.org/pocketsphinx>

Apart from human voice interaction, we have also tested sound source localization using HARK⁹ for possible people search when the person is outside of the robot visual perception area.

Iconic robot facial expression system. An iconic robot facial expression system as shown in Fig. 1 is under development. The iconic design is simple (low computing power requirement) yet expressive to create the character of the robot with more human-like expressions.

Cloud-connected. The robot system is controlled by an onboard computer system as the main robot controller to ensure stable low-level controls. Furthermore, the computer system can be connected with cloud systems for extra computing (e.g. image processing), knowledge database (e.g. dialogue engine) and online resources (e.g. wearable data).

3.4 Navigation in known and unknown environments (Help-me-carry)

With the top second vision system configuration (Fig. 6), we have developed the navigation system in known and unknown environments for *Help-me-carry* and *Restaurant* tasks. Based on the TurtleBot navigation package with prebuild map using *gmapping* and localization with *amcl* for waypoint travels, we has combined it with the people tracking package, for online update of the map while following the operator in the unknown environment. Taking an example scenario of the *Help-me-carry* task in Fig. 8, the robot is waiting at the initial point until someone is calling to follow her/him to a waypoint in the unknown environment, i.e. *car*. When reaching the car, the human will ask the robot to navigate back autonomously to a waypoint in the known environment, i.e. *kitchen*. Fig. 9 explains our program flowchart in solving the task.

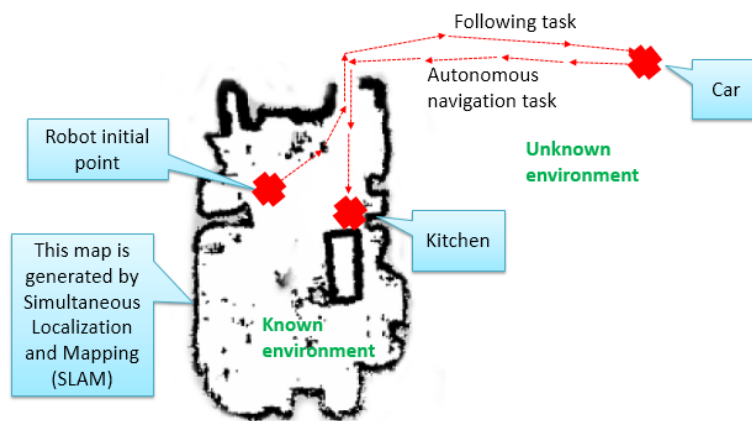


Fig. 8. Example scenario of the *Help-me-carry* task

⁹ <http://www.hark.jp/wiki.cgi?page=HARK-ROS-TURTLEBOT>

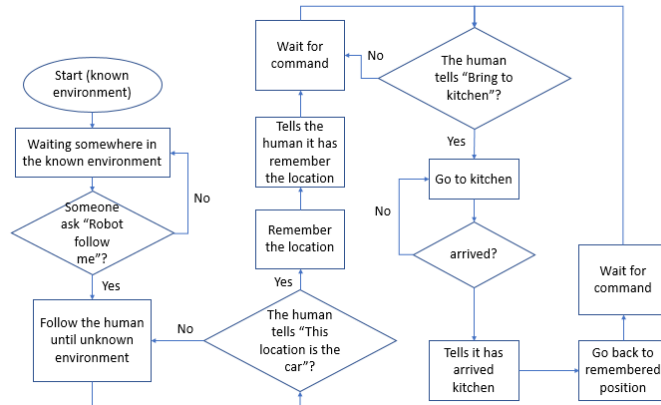
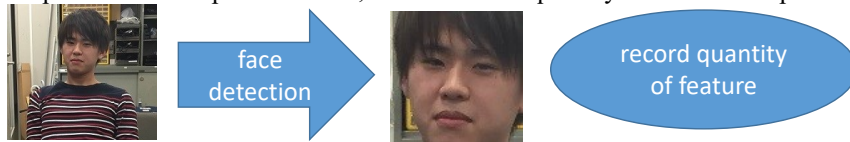


Fig. 9. The program flowchart of the *Help-me-carry* task

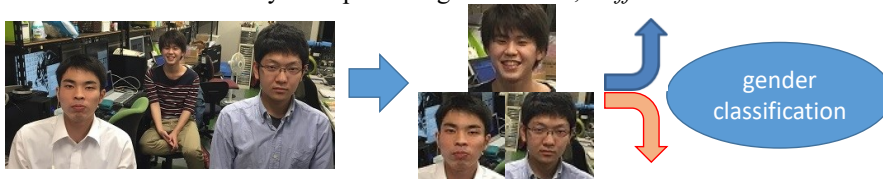
3.5 Person recognition system

On top of the basic robot platform, we have developed a module system for person recognition with the additional of a digital camera (Canon) with flash function and a laptop PC with NVIDIA 960GTX GPU. We use OpenCV image processing library and *Caffe* deep learning framework for the recognition. The algorithm is explained in Fig. 10.

Step1: Detect the operator's face, and record the quantity of feature in OpenCV



Step 2: Judge each person's gender of the crowd using the learning model generated by a deep learning framework, *Caffe*



Step 3: Search the operator from the crowd by matching the face feature using OpenCV



Fig. 10. Person recognition system

4 Scientific Contribution and Social Impacts

4.1 Open source robot platform for service robotics

This work aims to utilize open source solutions for both hardware and software to develop an open source robot platform for service robot research and development. The developed robot platform is open sourced with support wiki, source codes on GitHub and 3D printing parts to ensure easy reproducibility, to build up a community-driven development effort for service robots.

- Support wiki: <http://openbotics.org/kamerider/>
- Source codes (GitHub): <https://github.com/kamerider/>
- Demo videos (YouTube): <https://www.youtube.com/user/kameriderteam>

4.2 Participation in RoboCup competitions to benchmark the open source robot platform performance

On July 2015, we had participated for the first time in international RoboCup @Home in RoboCup 2015 Hefei, China with the open source robot platform (Fig. 11). With the overall ranked 7th result, it has proven the potential of the developed open source robot platform in RoboCup @Home application. Despite the increasing challenging of RoboCup @Home competition each year, we managed to maintain the 7th position in RoboCup 2016 Leipzig, Germany.

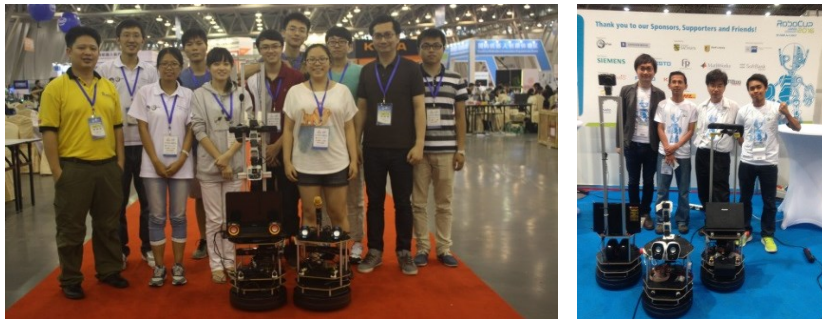


Fig. 11. RoboCup 2015 Hefei, China (left) and RoboCup 2016 Leipzig, Germany (right) Team KameRider is proud to be able to participate for the first time and ranked 7th overall with the open source robot platform in 2015, and maintained the 7th position in 2016

4.3 Support a new educational initiative – RoboCup@Home Education

RoboCup@Home Education¹⁰ initiative has been proposed in RoboCup Japan Committee in 2015. RoboCup@Home Education is an educational initiative in Ro-

¹⁰ <http://www.robocupathomeedu.org/>

RoboCup@Home that promotes educational efforts to boost RoboCup@Home participation and service robot development. Under this initiative, currently there are 3 projects started in Japan:

1. RoboCup@Home Education Challenge
2. Development of an educational open robot platform for RoboCup@Home
3. Outreach Programs (domestic workshops, international exchanges, etc.)

Along with this initiative, a series of related workshops (Fig. 12) and challenges, i.e. Intelligent Home Robotics Challenge 2014 and 2015 (Fig. 13) are taking place in Japan with the involvement of the open source robot platform.



Fig. 12. Workshop hosted by RoboCup@Home Education

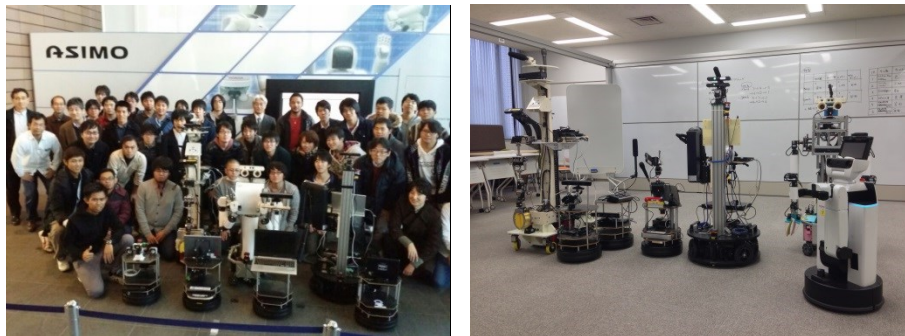


Fig. 13. Intelligent Home Robotics Challenge 2014 (left) and 2015 (right)

Team KameRider is proud to be the winner of Mobile Robot Category 3rd Place and Overall 3rd Place in the Intelligent Home Robotics Challenge 2014 with the open robot platform

A demo challenge (SPL Beta) of RoboCup@Home Education was organized in RoboCup Japan Open 2015 on May 2015 at Fukui, Japan (Fig. 14). A total of 6 new teams had participated in the challenge with encouraging performance.

Due to the great success of the demo challenge in RoboCup Japan Open 2015 and related developments (workshop and open source robot platform), RoboCup@Home Japan Committee has decided to continue this effort into an annual event in RoboCup Japan Open. In RoboCup Japan Open 2016 Aichi in March 2016 (Fig. 14), 9 teams

had participated in the education challenge, where 3 of them are qualified teams to take part in RoboCup 2016 Leipzig, Germany in July 2016.



Fig. 14. RoboCup@Home Education (SPL Beta) Challenge 2015 (left) and 2016 (right) KameRider team is proud to be the 1st place winner in 2015 and 2nd place winner in 2016 with the open robot platform

Not limited to domestic Japan, there are also RoboCup@Home Education themed international scientific exchange programs (Fig. 15) and internship programs, which are open for international participants.



Fig. 15. RoboCup@Home Education themed international scientific exchange programs for students from Malaysia (left) and China (right)

References

- [1] Thomas Wisspeintner, Tijn van der Zant, Luca Iocchi and Stefan Schiffer, "RoboCup@Home: Scientific Competition and Benchmarking for Domestic Service Robots", *Interaction Studies*, Vol.10, No.3 (2009), pp.392-426.
- [2] Tijn van der Zant and Luca Iocchi, "Robocup@Home: Adaptive Benchmarking of Robot Bodies and Minds", *Social Robotics*, (2011), pp.214-225.
- [3] Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.