KameRider SSPL @Home 2019 Team Description Paper

Jeffrey Too Chuan Tan[1]

[1] Nankai University, China
i@jeffreytan.org
http://openbotics.org/kamerider/

Abstract. This document is the team description paper of the KameRider SSPL team for the participation of @Home Social Standard Platform League in RoboCup 2019 Sydney, Australia. KameRider team is a collaborative effort that aims to promote educational robot development for service robotics. This paper describes the motivation of this effort, the development of SoftBank Robotics Pepper as standard hardware, software development, and the scientific contribution and social impacts of our work via the educational initiative – RoboCup@Home Education.

1 Introduction

KameRider team is a collaborative effort that aims to promote educational robot development for service robotics. Started from 2013, the very limited development resources and manpower team condition had urged a strong motivation to develop a more affordable yet functional educational robot solution to take part in RoboCup@Home league and service robot development. We are currently supporting the educational initiative of RoboCup@Home Education[1]. In this work, we use SoftBank Robotics Pepper robot [1] as Standard Platform to complement our educational purpose especially on the robot software and social interaction development.

2 Background and Motivation

2.1 The challenges of RoboCup @Home for Education

Starting from 2006, RoboCup@Home [2] 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 [3]. 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.

[1] http://www.robocupathomeedu.org/

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 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.

3 Robot Development for Education

3.1 Open source robot platform and new development of standard platform

The open robot platform we are developing serves as a fundamental robot platform for hardware and functional development. It has a current basic robot hardware configuration (Fig. 1) for fundamental robot platform and add-on modular component systems for customized applications. For example, 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.

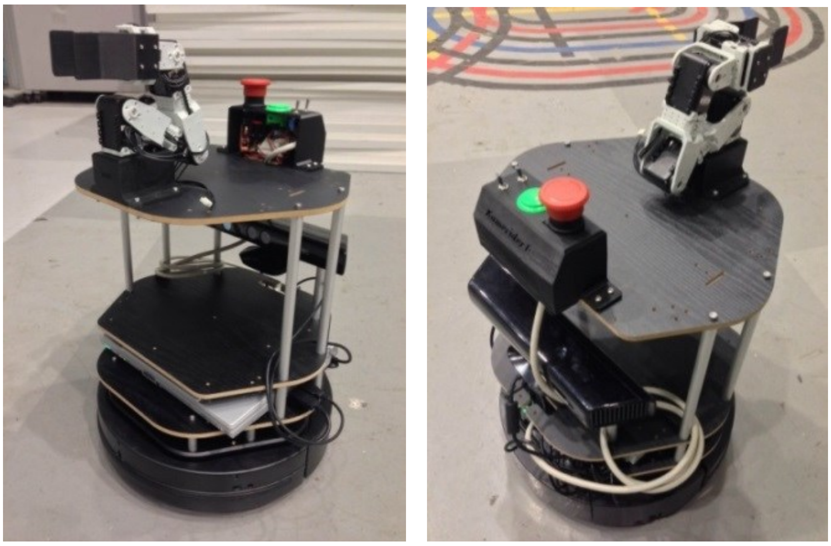

Fig. 1. Open robot platform for hardware and functional development

In this work, we use SoftBank Robotics Pepper[2] [1] robot as Standard Platform to complement our educational purpose especially on the robot software and social interaction development. Pepper is a commercially available robot platform that is specially designed to make interaction with human beings as natural and intuitive as possible. Pepper is capable of identifying the principal emotions by interpreting facial expressions such as smile, frown, and the tone of voice, as well as non-verbal language such as the movement of human head. The reliable hardware built and commercial grade technical support, plus the mentioned capabilities make it an ideal platform for our effort to develop educational platform that focuses on software and social interaction development.

3.2 ROS as the robot software framework with cloud system

ROS (Robot Operating System) [4] 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, our development is adapting and assembling ROS packages and stacks to realize the navigation, manipulation, vision and speech functions of the robot in order to perform the service robotics tasks in RoboCup@Home. Pepper robot uses Aldebaran's NaoQI framework. We use NaoQI API in ROS via the *naoqi_driver* to integrate our current system into Pepper. We are currently studying the Pepper robot hardware (Fig. 2) [1], sensors and sensing ranges (Fig. 3) [1] to correspond with our open platform robot system for development.

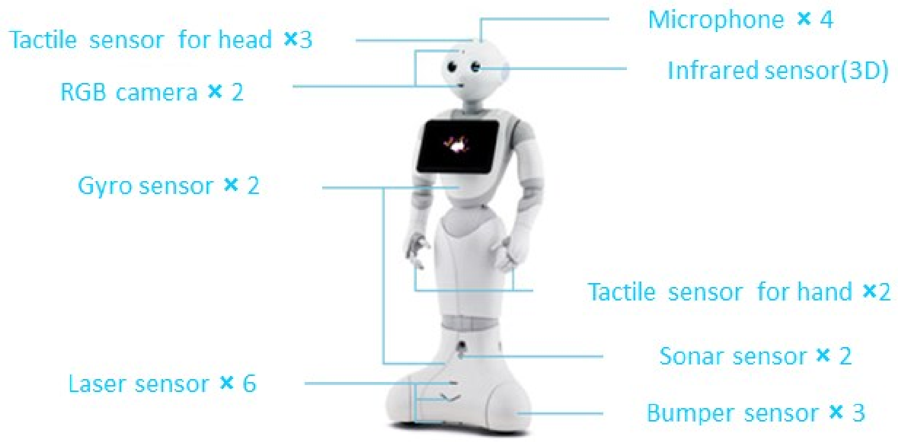

Fig. 2. Pepper robot hardware [1]

[2] https://www.ald.softbankrobotics.com/en/cool-robots/pepper/find-out-more-about-pepper

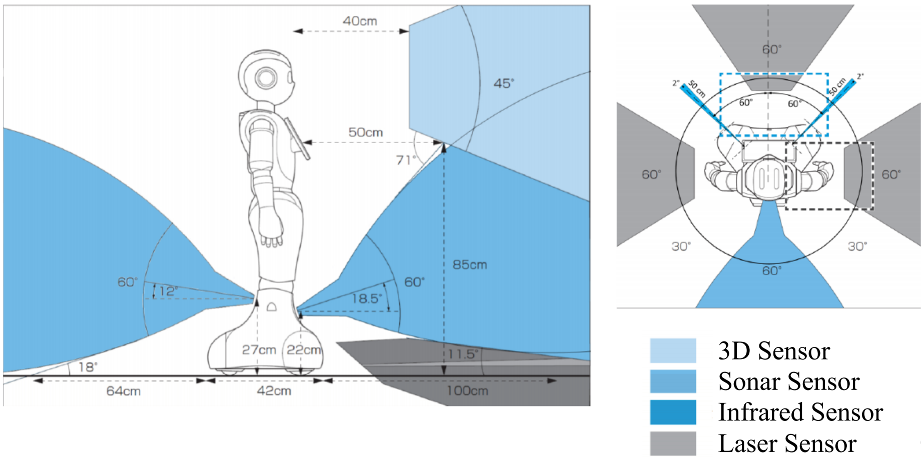

Fig. 3. Pepper robot sensors and sensing ranges [1]

To address the low onboard computing capabilities in our robot development, we are constructing connected robot platform with cloud system for extra computing (e.g. deep learning for image processing), knowledge database (e.g. dialogue engine) and online resources (e.g. wearable data). This cloud system will be useful for Pepper robot as standard platform to expand its capabilities without any modifications on the hardware.

4 Scientific Contribution and Social Impacts

4.1 Open source robot development for service robotics

We have participated in the first @Home SSPL competition in RoboCup 2017 Nagoya, with the final ranking of 4th place. As we aim to promote educational robot development for service robotics, we open source our development with support wiki and source codes on GitHub for sharing, with the aim to build up a community-driven development effort for service robots.

- Website: **http://openbotics.org/kamerider/**
- Learning resources: **http://www.robocupathomeedu.org/learn**
- Source codes: **https://github.com/robocupathomeedu/**
- Demo Videos: **https://www.youtube.com/user/kameriderteam**

4.2 Robot navigation

Pepper has its own laser sensors which are located on the bottom of the robot, but we can get very limited data from them, because of the small coverage angle. So we decided to generate a laser through the RGB-D camera's point cloud (like the white line in Fig. 4 below).

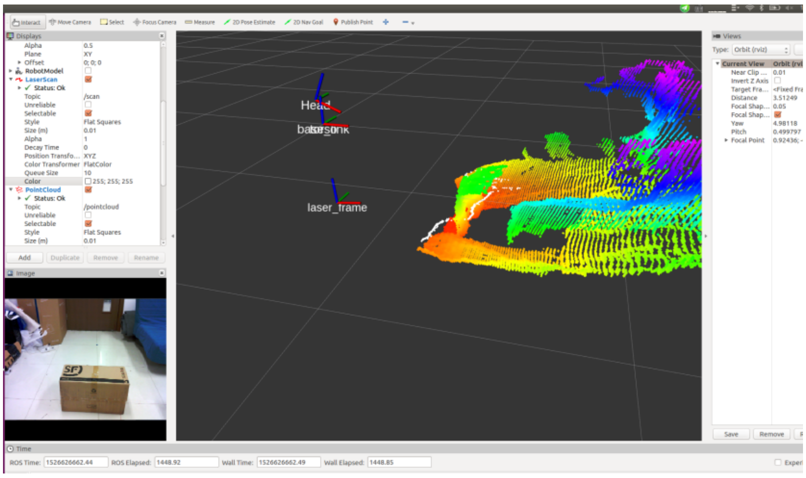

Fig. 4. Sensor data visualization in Rviz

Thus, we can use the data of these two types of laser sensors in the pepper's navigation. In Fig. 5 below, the yellow color represents pepper's own laser sensor data and the red color represents the laser generated from RGB-D camera.

ROS navigation package[3] 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*[4].

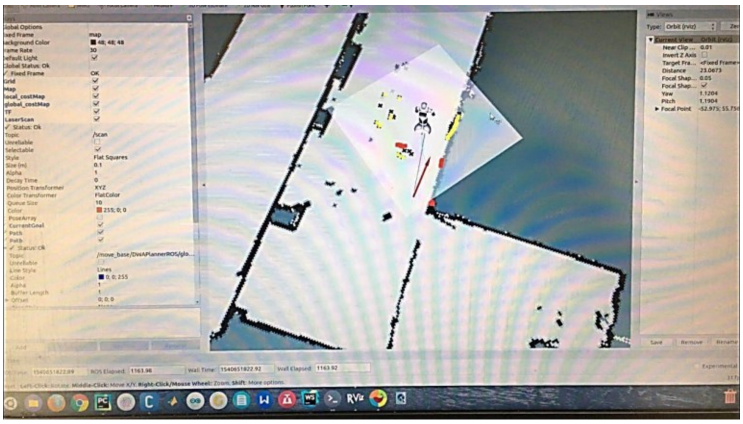

Fig. 5. Pepper navigation system

[3] http://wiki.ros.org/turtlebot_navigation/

[4] http://wiki.ros.org/navigation/Tutorials/SendingSimpleGoals

4.3 Robot vision and person/gender/object recognition system

Person recognition. We built a Convolutional Neural Network (CNN) with Tensorflow as well as Keras. When we need to add a new person into our database, we use a digital camera to take about 1000 pictures of this person, and then we use Haar-like face detection method to detect human face in each picture and add the face region of each picture to our database with its labels. After that we use a laptop with NVIDIA GTX1070 to run the CNN, so that we can get our own model to recognize the person. The algorithm is explained in the following figure.

Gender recognition using an online API from Baidu-AI. During the competition, we capture and upload the photo to the Baidu-AI cloud server, and we can get the gender recognition results labeled on the photo.

Object recognition system. We use YOLO (You Only Look Once) for object detection. In the *Storing Groceries* task, we use the Kinect sensor for shelf detection, table detection and object detection. Before the competition, we take photos of the predefined object, and then we do labeling by adding annotation labels and bounding boxes for each image. We capture the images in different angle, different light condition as well as different background to ensure suitable generalization of our model to deal with the competition conditions.

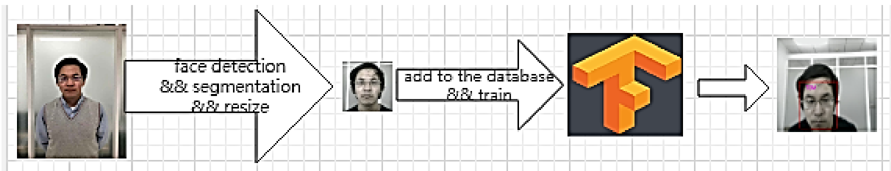

Fig. 6. Person recognition system

Human gesture detection. In our human gesture detection system (Fig. 7), we use CMU OpenPose as our skeleton detector (Fig. 8). It is a real-time multi-person keypoint detection library for body, face, hands, and foot estimation. The OpenPose demo requires a RGB image and then returns the number of people as well as their skeleton positions. To get the human pointing direction, 3D coordinates of the wrist joint and elbow joint are necessary. We combine the OpenPose result and point cloud library (PCL) to get the positions described in the head RGB-D sensor coordinate system, then the TF matrix is used to transform them to the map coordinate system. Additionally, space vector method is used to calculate which point on the ground the human is pointing to.

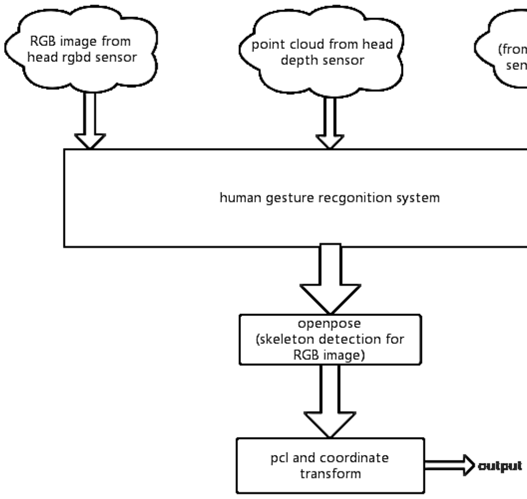

Fig. 7. Human gesture detection system

Fig. 8. CMU OpenPose skeleton detector

4.4 Simulation Development with SIGVerse

In order to speed up our robot development, we are also developing robot simulation with SIGVerse[5]. SIGVerse is a robotics simulator that can simulate human-in-the-loop human-robot interaction, capable of representing various task scenarios in RoboCup @Home.

Refer to Fig. 9, we have developed a virtual Handyman task that resembles the GPSR task in @Home. We develop and improve our gesture detection system in the virtual Interactive Clean Up task for better human gesture recognition. In the virtual human navigation task, we train our robot system to understand the sentence generation in GPSR. We also use the power of simulation system to conduct repetitive robot learning of huge amount of data via crowdsourcing.

[5] http://www.sigverse.org/

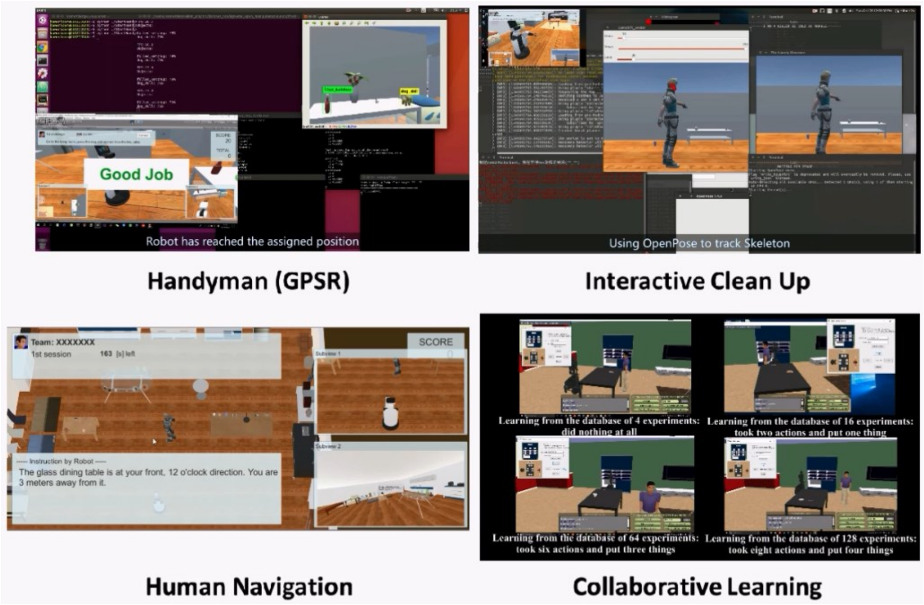

Fig. 9. Simulation development with SIGVerse

References

[1] SoftBank Robotics Pepper robot, https://www.softbank.jp/en/robot/

[2] 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.

[3] Tijn van der Zant and Luca Iocchi, “Robocup@Home: Adaptive Benchmarking of Robot Bodies and Minds”, *Social Robotics*, (2011), pp.214-225.

[4] 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.

Annex

Name of team: KameRider SSPL
Contact Information: i@jeffreytan.org
Website: http://openbotics.org/kamerider/

Hardware:

- SoftBank Robotics Pepper robot

Software:

- ROS (Robot Operating System)
- CMU Pocket Sphinx
- OpenCV
- Baidu-AI
- Tensorflow and Keras
- OpenPose