

Team Tidyboy RoboCup@Home Domestic Standard Platform League Team Description Paper

Seung-Joon Yi¹, Chung-Yeon Lee², Jaebong Yi¹, Hyunjoon Cho³,
Youngbin Park⁴, Byoung-Tak Zhang², Jae-bok Song³, Il Hong Suh⁴

¹Department of Electrical Engineering, Pusan National University, Busan, Korea

²Department of Computer Science, Seoul National University, Seoul, Korea

³Department of Mechanical Engineering, Korea University, Seoul, Korea

⁴Department of Electronics Engineering, Hanyang University, Seoul, Korea

Corresponding email to: seungjoon.yi@pusan.ac.kr

Abstract. Team Tidyboy is a RoboCup@Home Domestic Standard Platform League (DSPL) team that consists of members from Pusan National University, Seoul National University, Korea University and Hanyang University. We have previously participated in two robotic competitions using the Toyota Human Support Robot (HSR) platform with promising results, RoboCup@Home DSPL 2018 and World Robot Summit (WRS) 2018, and also have extensive expertise in other robotic competitions including RoboCup soccer leagues, DARPA Robotics Challenge (DRC) and RoboCup@Home Social Standard Platform League (SSPL). In addition, we have a strong research experience in state-of-the-art machine learning methods. In this paper, we present our software framework for the HSR platform, and how we will prepare for the upcoming RoboCup 2019 with help of newly developed software modules such as socially-aware navigation, visual question-answering, and schedule learning.

1 Introduction

Team Tidyboy is a joint RoboCup@Home DSPL team that consists of members from Pusan National University, Seoul National University, Korea University and Hanyang University. We have participated in two recent international robotic competitions using the Toyota HSR platform, RoboCup@Home DSPL 2018 held in Montreal, Canada and WRS 2018 held in Tokyo, Japan. We also have extensive expertise with other robotic platforms as well, including RoboCup soccer leagues, DRC and RoboCup@Home SSPL. In addition, we have a strong research experience in state-of-the-art machine learning methods applied to various robotics problems. In this paper, we present our software framework we have used for recent robotic competitions using the HSR platform, and describe how we can improve our code for the upcoming RoboCup@Home DSPL 2019, with help of newly developed software modules such as socially-aware navigation, visual question-answering, and schedule learning.

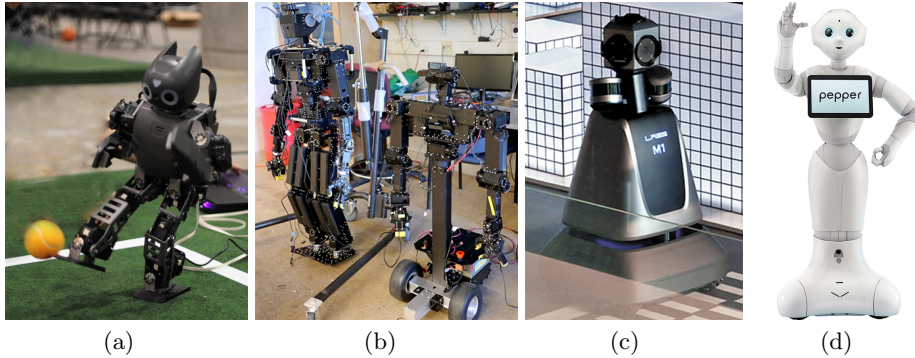


Fig. 1. Robotic platforms we have previously worked on. (a) Robotis DARwIn-OP (b) Robotis THOR-OP (c) Naver Labs M1 (d) Softbank Pepper

2 Hardware

Our team currently has two HSR platforms, which are generously provided by Toyota corporation for RoboCup@Home and World Robot Summit competitions. To be able to test concurrently without physically meeting together, we keep them separately in two universities. In addition to the HSR platforms, we have worked with a number of other robotic platforms in highly competitive environment. Here we introduce the robots we have previously worked on, and show how the previous experience of using the robots have helped us rapidly develop the software for the HSR platform.

2.1 DARwIn-OP Soccer Robot

DARwIn-OP is a 45 cm tall miniature humanoid robot designed primarily for RoboCup Humanoid KidSize League [5]. It has two legs with six degree of freedom (DOF) each for bipedal locomotion, and two 3DOF arms mainly used for getting up after a fall. A single RGB camera in the head is used for perception and an Inertial Measurement Unit (IMU) in the torso is used for balancing. During the match, the robot operates in full autonomy. Due to the nature of the competition, the robot has to make quick, real-time decisions to outmaneuver the opponents - which has been the main focus of our high level behavior logic.

2.2 THOR-OP Hazardous Rescue Robot

THOR-OP is an 1.47 m tall humanoid robot designed for the DRC competitions, which pose a number of difficult mobility and manipulation tasks such as driving a car, climbing a ladder and using power tools. The robot is teleoperated, but the competition still requires autonomy due to the throttled communication. It has two 6DOF legs for locomotion, two 7DOF arms with grippers for precise manipulation, and a 2DOF waist that helps expanding the workspace. As the competition requires precise mobile manipulation capability, we have developed

a hierarchical, task-specific arm motion library and planner - which we use for the arm motion generation for HSR platform as well.

2.3 M1 Autonomous Indoor Mapping Robot

M1 is an omnidirectional wheeled robot developed by Naver Co. Ltd. in Korea, with the goal of autonomous exploration and generation of a high-resolution 3D textured map of indoor space. It has Mecanum wheels for omnidirectional mobility, three Velodyne multi-channel LIDARs for high-resolution depth mapping and a Ladybug spherical camera system for recording spherical image. The robot has demonstrated its autonomous exploration and mapping capability at the Seoul Motor Show 2017 held in Seoul, Korea [3]. We plan to migrate the mapping and localization module developed for this robot to the HSR platform.

2.4 Pepper Indoor Service robot

Pepper is the standard platform for RoboCup@Home SSPL. The robot has a omnidirectional drivetrain and two 5DOF arms that can be used for object manipulation and gesture based human robot interaction. It has a number of sensors including a Xtion RGBD camera, 2 RGB cameras, 4 microphone array, 6 laser range sensors and 2 ultrasonic sensors. Pepper robot was used for the RoboCup@Home SSPL 2017 and 2018 leagues by our SSPL team, team AU-PAIR, showing advanced perception and human-robot interaction capabilities. We plan to migrate the codes to the HSR platform for better perception and situational awareness capability for the RoboCup@Home DSPL 2019.

3 Software

3.1 Overall Architecture

Our software framework has its roots in the RoboCup humanoid league [8]. It is designed to be highly modular to support a variety of robotic hardware and be quickly ported on new robot platforms with minimal effort, as well as various robotic simulators. We also have the ZeroMQ messaging and shared memory layout for inter-device and inter-process communication. Although our custom framework can completely replace the ROS framework HSR platform uses, we have decided to keep both for quick development and easy debugging. The external computing device communicates with the robot by ROS messages, and we run ROS message handler in the external device that converts between internal shared memory data and ROS messages.

3.2 SLAM

For indoor mapping and localization, we currently use the hector-slam and amcl packages. Those packages generally work well in many environments but they

require a pre-built map and we have seen frequent localization failure cases under some specific scenarios, for example when the robot opens a cabinet drawer by whole body movement. We plan to substitute the mapping and localization module with our Iterative Closest Point (ICP) based 3D SLAM algorithm, which can incrementally generate the traversability and frontier map on the fly for autonomous navigation.

3.3 Navigation

For the RoboCup@DSPL 2018 competition, we mainly used the ROS navigation stack to move the robot around. However, the default navigation package has many issues for indoor navigation - it is fairly slow, and very sensitive to possible dynamic obstacle observed by the head RGBD camera, which can sometimes make the robot stuck and fail to move, which happened to our team during the grocery task of RoboCup 2018. So for the WRS 2018 competition we let the robot navigate to relatively open space first using the ROS navigation stack, and then move the robot close to the manipulation target using velocity control while ignoring the dynamic obstacles. Still we have found that the navigation is often the slowest link of the whole behavior chain, and the robot stops moving for far too long when nearby obstacle is detected. We plan to completely replace the navigation code with our own code, with potential field based continuous obstacle avoidance.

3.4 Manipulation

The HSR platform has a limited degrees of freedom for its manipulator, so a general purpose arm motion planner cannot be used without utilizing the base movement. Instead of using a general purpose arm planner, we use a library of parameterized arm motions to handle objects at various heights and locations. We have total of 5 different arm motions that can reach manipulation target from the ground to 1.1 meter high, pick up postcard using the suction nozzle, as well as pick up very small objects such as forks and spoons. In addition, we have made whole body motion library to manipulate objects such as refrigerator door or cabinet shelf. To increase the chance of picking up very small objects, we devised a progressive grasping motion that advances the gripper position while gripping, keeping the end tip of the gripper at the same height. With help of force sensor feedback and this progressive grasping motion, the robot can pick up small objects on the surface with high probability even if the position estimate is a few centimeters off.

3.5 Communication

HSR provides a good text-to-speech (TTS) module for voice synthesis. After testing various speech recognition APIs, we have decided to use the Google Cloud Speech Recognition API that gave us the best result. The google API



Fig. 2. HSR manipulating various objects at World Robot Summit 2018 competition.

gave us a very good result in the RoboCup@Home 2018 DSPL competition, but it tends to give wrong recognition results when a non-native English speaker provides the voice. We plan to build a task specific heterograph library to handle such issues.

3.6 Perception

For autonomous indoor service tasks required for RoboCup@Home leagues, a robust perception capability utilizing multiple onboard sensors is crucial. Our current object detection pipeline first uses the YOLOv3 [9] model trained using the actual object image for the competition, and then uses the detected object bounding boxes to get per-object point cloud from matching depth image. To filter the point cloud, we use various information such as object storage height candidates and object geometry, and the filtered point cloud is clustered by k-nearest neighbor algorithm. After clustering, we run the principal component analysis (PCA) algorithm to get the correct grasp pose for the object. For human detection, we use Kairos online API [2] that can detect human faces and determine their gender, race, age and other attributes from RGB image. For up-

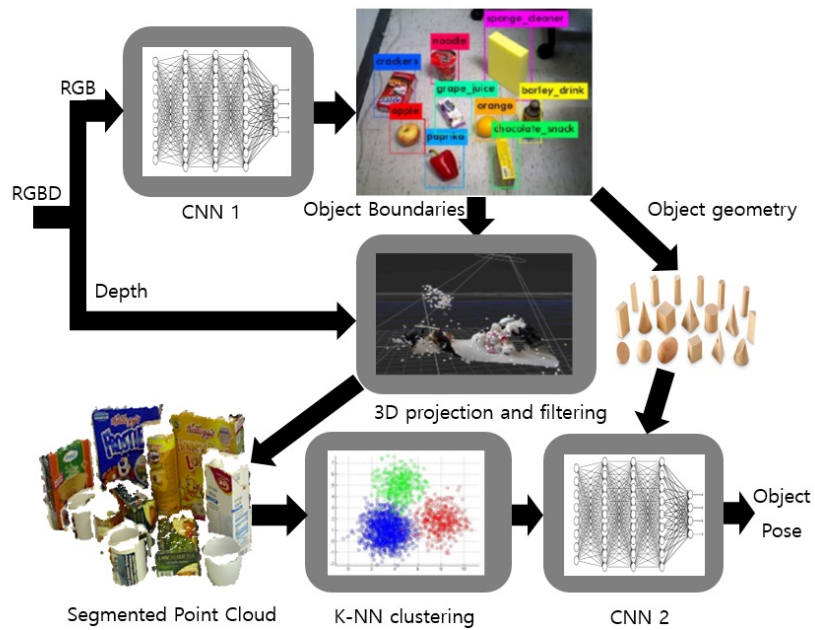


Fig. 3. Perception structure

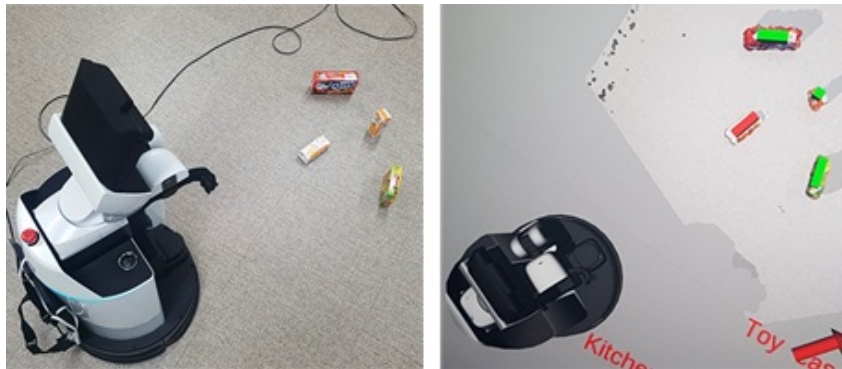


Fig. 4. HSR getting the grasp pose of the objects on the ground

coming RoboCup, we plan to use the human pose detector such as OpenPose [4] as well to detect human in various posture.

3.7 Autonomy

RoboCup soccer league requires a complete autonomy for a team of robots in a dynamic and adversarial environment. In our framework, the autonomous behav-

ior is handled by maintaining a number of parameterized finite state machines (FSMs) running in parallel. The autonomy is extensively tested and optimized through repeated self-play trials in simulated environment utilizing reinforcement learning algorithm. In addition to this FSM based architecture, we have added a task queue structure that can queue a number of actions and execute them sequentially. We have used the task queue architecture in WRS 2018 competition, where the robot has successfully executed complex high order commands which consist of more than 10 sequential tasks.

4 Conclusion

Having a proven background in developing successful robot systems, especially in front of international audience of RoboCup, WRS and DRC competitions, team Tidyboy vows to further service robot research in localization, navigation, manipulation, perception and human robot interaction by competing to its best abilities in upcoming RoboCup in Sydney. We have open sourced our RoboCup humanoid soccer software, which has been widely adopted by a number of teams, as well as the software and dataset used for RoboCup@Home DSPL 2018. We wish to contribute to the RoboCup@Home league as well by releasing our codes and data after the competition.

References

1. IPSRO integrated perception framework, <https://github.com/gliese581gg/IPSRO>
2. Kairos face detection api, <https://www.kairos.com/>
3. Naver's self-driving robot highlights future ambitions (2017), <http://koreabizwire.com/navers-self-driving-robot-highlights-future-ambitions/79277>
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
5. Ha, I., Tamura, Y., Asama, H., Han, J., Hong, D.W.: Development of open humanoid platform darwin-op. In: SICE Annual Conference 2011. pp. 2178–2181 (2011)
6. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4565–4574 (2016)
7. McGill, S.G., Yi, S.J., Lee, D.D.: Low dimensional human preference tracking for motion optimization. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 2867–2872 (May 2016)
8. McGill, S.G., Brindza, J., Yi, S.J., Lee, D.D.: Unified humanoid robotics software platform. In: The 5th Workshop on Humanoid Soccer Robots (2010)
9. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
10. Yi, S.J., McGill, S., Hong, D., Lee, D.: Hierarchical motion control for a team of humanoid soccer robots. International Journal of Advanced Robotic Systems 13(1), 32 (2016)

HSR Software and External Devices

We use a standard HSR robot from Toyota. No modifications have been applied.



Fig. 5. Toyota HSR

Robot's Software Description

For our robot we are using the following software:

- OS: Ubuntu 16.04
- Middleware: ROS Kinetic and in-house codebase
- Localization and Mapping: ICP and particle filter based in-house algorithm
- Arm control: In-house arm motion planner [7]
- Navigation: In-house hierarchical motion planner [10]
- Integrated recognition: IPSRO [1]
- Object recognition: YOLOv3 [9]
- Pose estimation: OpenPose [4]
- Image Captioning: DenseCap [6]

External Devices

Our robot relies on the following external hardware:

- Official Standard Laptop: Intel i7 CPU, 32GB RAM, NVIDIA 1080 GPU
- External Computing Device: Intel i7 CPU, 32GB RAM, NVIDIA Titan XP GPU

Cloud Services

Our robot connects the following cloud services:

- Speech recognition: Google Cloud API
- Image recognition: Kairos API