In *Proceedings of the International Joint Conference on Artificial Intelligence* (IJCAI), New York City, USA July 2016

Robot Scavenger Hunt: A Standardized Framework for Evaluating Intelligent Mobile Robots

Shiqi Zhang[1], **Dongcai Lu**[2], **Xiaoping Chen**[2] and **Peter Stone**[1]
[1] Department of Computer Science, University of Texas at Austin
{szhang,pstone}@cs.utexas.edu
[2] Computer School, University of Science and Technology of China
ludc@mail.ustc.edu.cn, xpchen@ustc.edu.cn

1 Motivation

In recent years, many different types of intelligent mobile robots have been developed in research and industrial labs. Although there are significant differences in both hardware and software over these robots, many of them share a common set of AI capabilities, e.g., planning, learning, vision and natural language processing. At the same time, almost all of them are equipped with traditional robotic capabilities such as mapping, localization, and navigation. However, to date it has been difficult to compare and contrast their capabilities in any controlled way. The main goal of the *Robot Scavenger Hunt* is to provide a standardized framework that includes a set of standardized tasks for evaluating the AI and robotic capabilities of medium-sized intelligent mobile robots. Compared to existing benchmarks, e.g., *RoboCup@Home*[1], Robot Scavenger Hunt aims at evaluations in larger spaces (multi-floor buildings vs. rooms) over longer periods of time (hours vs. minutes) while interacting with real human residents.

2 Game Rules

A *task library* that includes a set of standardized tasks plays the key role in the Robot Scavenger Hunt. Individual tasks in the task library will be described in the next section. Robot players can participate in the hunt in their local buildings, but the players have to upload *completion certificates* (through cloud file hosting service) as evidence of task completion. It is not mandatory for the robot to work on all tasks in the task library when working on the tasks. Instead, we encourage robots to dynamically adjust the order of tasks in such a way that they judge will maximize their overall score. For instance, tasks requiring long-distance navigation can be executed at times when humans have fewer activities, e.g., early mornings, while tasks require human-robot interactions can be attempted during hours when people are more accessible. However, this reordering should be done fully autonomously: once the robot starts to work on the tasks, its only input from a person should be as specified by the tasks themselves. Specific tasks can be conducted more than once or not at all in each life cycle. To prevent a robot player from repeatedly working on the same (easy) task continually, we set a *minimum time gap* for each task (for now, it is 1 hour for all tasks). A task completion counts, only if the gap between the last completion of this task and the current one is larger than its minimum time gap.

Other than the task library, the Robot Scavenger Hunt has an *object library*. This object library includes a set of object names that each corresponds to a class of small-size objects that are easily accessible from most countries in the world. The main reason of introducing this object library is to scope the tasks that require object manipulation and make it possible to compare robots that work in very different environments. Example objects in this object library include tea cups, coke cans, and bowls.

A task specification is a four-tuple: $\langle$*task name*, *certificate format*, *score*, *description*$\rangle$. The *task name* is a unique string used for identifying the task. The *certificate format* specifies the format of completion certificate. For instance, tasks that require navigation will have certificate formats that include an image showing the robot's traveled trajectory printed on a map. The *score* defines the reward a robot can receive when the certificate meets the requirements specified in the task description. Finally, in the *description*, we describe the requirements that a completion certificate needs to meet so the robot can earn the full score, and in case of partial completion or partial correctness, how we award partial credit.

3 Task Library

The task library of Robot Scavenger Hunt includes a set of tasks that are designed in such a way that: 1) each task demonstrates at least one AI capability and one robotic capability that are listed in Section 1; and 2) a competent robot can succeed most of the time given sufficient time in a standard building environment. Following this strategy, we initially add the four tasks below into the task library.

Color shirt In this task, the robot needs to take a picture of a person who wears a shirt of a specific color (such as red, green, and blue). The robot can decide to either stand still to detect people (and their shirt colors) in a passive way, or actively move to scenes to find such people. This task aims at evaluating a robot's vision and navigation capabilities. The full score of this task is 100. If there is a person in the picture but the shirt color does not match, we give one third of the full score. Otherwise, the robot receives a zero score.

Human following A human following task requires a robot to physically follow a human for at least 10 meters. Similar

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

to “color shirt”, this task is used for evaluating navigation and vision capabilities, while emphasizing more on navigation. As completion certificates, the robot needs to save an image that prints, in different colors, the traveled trajectories of both the human (who is followed by the robot) and the robot itself. At the same time, it is required to save at least one picture that includes the human being followed by the robot. This task has a full score of 100 and has no partial score.

Object delivery This task requires a robot to move an object from one room to another. This task aims at evaluating robots’ manipulation and planning capabilities. However, a robot can choose to ask for human help instead of loading and unloading the object using robot arm. In that case, the robot will need human-robot interaction (HRI) capabilities such as natural language processing (NLP) and gesture recognition, e.g., for pointing to the object that needs to be loaded. As completion certificates, the robot has to report the traveled trajectory in working on this task and at least two images showing the object’s positions before and after the delivery. The full score of this task is 150.

Target search A target search task is specified by an object name selected from the *object library*. Each robot can choose a format that works the best for its object recognition algorithm (e.g., RGB-D visual features). In doing this task, the robot has to physically move from point to point to visually analyze different scenes in the environment. Good reasoning capabilities can guide the robot to search areas where the target most likely is while balancing navigation costs. At the same time, robot players can actively seek human help as needed. This task’s full score is 200. Obviously, it is not allowed to continuously “search” for the same object, even if the time gap meets the requirement of minimum time gap (introduced in Section 2).

4 Robot Participants

To date, the following two robots have participated in the scavenger hunt.

KeJia Robot The hardware of KeJia robot is shown in Figure 1 (**Left**). Our robot is based on a two-wheels driving chassis of 62*53*32 centimeters in order to move across narrow passages. Its sensors include a laser range finder, a 1394 camera and a kinect. A lifting system is mounted on the chassis attached with the robot’s upper body. Assembled with the upper body is a five degrees-of-freedom (DOF) arm. It is able to reach objects over 83 centimeters far from mounting point and the maximum payload is about 500 grams when fully stretched. The robot’s power is supplied by a 20Ah battery which guarantees the robot a continuous running of at least one hour. The computational resources consist of a laptop and a on-board PC.

We have created an integrated architecture for our robot. In general service scenarios, our robot is driven by human speech orders, as input of the robot’s Human-Robot Dialogue module. Through the Language Understanding module, the utterances from users are translated into the internal representations of the robot. The Task Planning module then generates a low-level plans for these representations. The

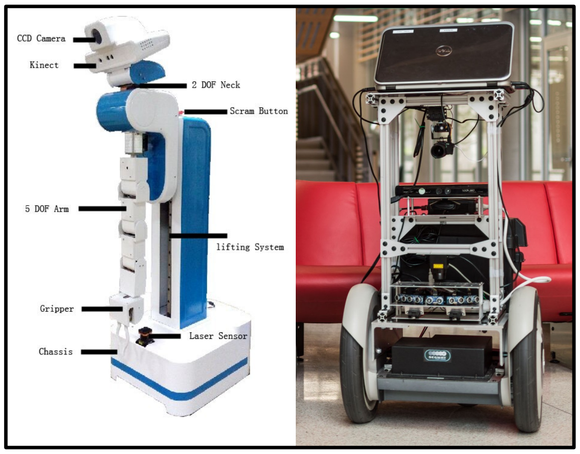

Figure 1: Robot players in the Robot Scavenger Hunt game: Robot KeJia D2 (**Left**) and BWIBot (**Right**).

generated course of actions is fed into the Motion Planning module. Each action is designed as a primitive for KeJia’s Task Planning module and could be carried out by the Motion Planning module and then autonomously executed by the Hardware Control module. A video of a KeJia robot participating in the robot scavenger hunt can be viewed online: `https://youtu.be/yIwDeM7p7Jc`

BWIBots Figure 1 (**Right**) shows the other robot platform that has been used in the Robot Scavenger Hunt. BWIBots are built on the differential drive Segway RMP 50. The Segway base allows up to $2m/s$ speed, while we limit it to $1m/s$ for safety concerns. Each BWIBot has a lead-acid battery (12V) for powering both the Segway base and peripherals such as computer and sensors. Onboard computation is on a laptop with an Intel i7 processor. BWIBots are equipped with sensors that have complementary features for localization, obstacle avoidance, and human robot interaction, including a 2D planar LIDAR mainly used for building 2D occupancy-grid maps and localization and a Microsoft Kinect mostly used for HRI and obstacle avoidance. One of the five BWIBots has a Kinova Mico 6-DOF arm for manipulation tasks.

The software of BWIBots is built on top of the Robot Operating System (ROS) middleware [Quigley *et al.*, 2009]. In order to navigate safely and autonomously in a multi-floor environment, each BWIBot has a map server that provides all maps needed to perform navigation across all floors. Above the level of motion control, all BWIBots have a symbolic planning level for computing a sequence of actions to achieve goals that are unreachable through individual actions. A video of a BWIBot working on scavenger hunt tasks can be viewed online: `https://youtu.be/l8zh0nwzedw`

References

[Quigley *et al.*, 2009] Morgan Quigley, 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*, 2009.