

Toward a robot that acquires logical recognition of space

Megumi Fujita^{*}, Yuki Goto[†], Naoyuki Nide[‡],
Ken Satoh[§], Hiroshi Hosobe[¶]

Abstract

For cooperation between robots and humans, a robot should have logical recognition regarding space. For example, if a robot, such as a housekeeping robot, can recognize the arrangement structure of the furniture and the concept of “a room”, this information will be useful for asking the robot to carry out tasks. Therefore, our aim is for the robot to acquire information on the relationship between locations while moving. So, in hopes that the robot can recognize that “I have exited the room”, we conducted an initial-stage experiment involving a robot leaving a room using knowledge of the relationship between the entrance of the room and the door. This paper describes the insights obtained from this experiment.

Keywords: Autonomous robots, Situation recognition, Logical inference of actions

1 Introduction

In the past, as research on the means for robots to move in indoor or outdoor spaces, technologies, such as spatial recognition by SLAM and generation of moving plans using planning, have been developed. However, these technologies lack the concept of logical recognition of the spaces in which robots are located.

For example, suppose that a human says to a robot that “please put this plushie over the cabinet in the next room”. At that time, if the robot does not logically understand the relationships between the rooms or the furniture arrangement in the room, the robot cannot understand the “next room”, and it is difficult for the robot and the person who asked for that task to communicate with each other.

From such a point of view, it is necessary to acquire the logical recognition of a space and maintain it as knowledge for robots, such as housekeeping robots, to coexist with humans. If a robot has abilities such as recognizing the furniture arrangement in a room structurally, and obtain the concept of “a room” as a single place, the robot will be able to recognize from which room to which room it has moved. If the robot can gain such recog-

^{*} Graduate School of Humanities and Sciences, Nara Women’s University

[†] Graduate School of System Informatics, Kobe University

[‡] Faculty of Human Life and Environment, Nara Women’s University

[§] Principles of Informatics Research Division, National Institute of Informatics

[¶] Faculty of Computer and Information Sciences, Hosei University

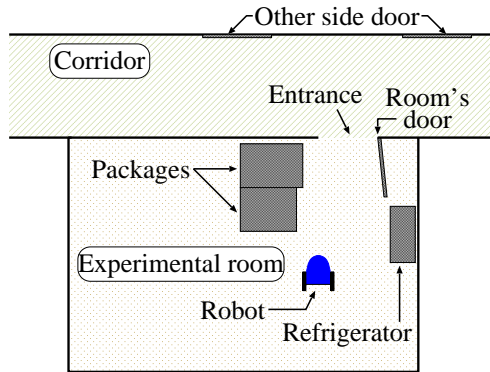


Figure 1: Robot's environment in experimental room



Figure 2: Q.bo Lite Evo

dition and keep it as knowledge, the information will be useful for asking the robot to carry out tasks.

From this viewpoint, we conducted a verification experiment in which the robot was given a relationship between a door and the entrance in a room as knowledge, then it exited the room through the entrance using that knowledge (Fig. 1). In this experiment, we aimed for the robot to recognize that it has got out of the room. However, due to various problems, our aim has not been achieved yet at present. In this paper, we describe the findings we gained from the experiment and discuss them.¹

Note that this paper focuses on the discussion on the problems for robots to acquire logical recognition regarding places. Our larger goal is for robots to discover new knowledge about the relationships between rooms using relationships obtained from its observations. For discussion from that point of view, please refer to our another study[1].

2 Implementation of our robot program

2.1 Robot and BDI model

For this study, we used Thecorpora's Q.bo Lite Evo robot (Fig. 2). It has cameras as its eyes, which can capture the forward view. It also has two ultrasonic sensors near its front wheel that can detect obstacles.

Our implementation of the robot's behavior is based on the BDI model[2], a model of autonomous agents that emulates the process of human goal achievement. A BDI agent, an agent based on the BDI model, first generates its goal using its beliefs obtained from its environments. By practical reasoning[3], using these beliefs, it selects a means (basically from its plan library) to achieve the goal, forms it as its intention, then attempts to maintain the intention and execute it (the large box at the top of Fig. 3).

To make the robot have a goal and act while reasoning its sub-goals and plans to achieve them, we used the BDI model for the robot's action decision. To implement this, we used Jason[4], a platform for implementing the BDI architecture (the fundamental agent architecture on the BDI model). With Jason, practical reasoning can be described in a logic programming language approximately corresponding to Prolog.

¹The video of this experiment is available at <http://blackknight.ics.nara-wu.ac.jp/~kuma/index2.html#roboexit> as of December 10, 2017.

Although the robot’s goals and sub-goals are simple, the use of the BDI architecture has a clear advantage that we can describe goal-oriented actions in a Prolog-like language. In addition, to achieve our future goal of actualizing robots that attempt to reach their destinations by forming sub-goals, we can take advantage of the BDI model in which the robot can hold or modify its intentions and maintain its sub-goals in response to environmental changes or failures of actions in the real world[5].

2.2 ROS and Jason

The robot’s motors and sensors are controlled using ROS[6]. We use ROS from Python (Fig. 3). By processing the visual information from the sensors using libSVM, the (non-)presence of a target object is determined, and the result is passed to Jason. Jason then acquires the result as a perception and uses it for practical reasoning to determine the robot’s action, i.e., selects the plan for the current goal and passes an atomic action to the Python side. On the Python side, the motor output determined based on that action is passed to ROS, and the robot acts.

We are currently focused on developing a mechanism for determining actions from goals as a practical reasoning based on logic programming. Hence, object recognition using libSVM currently involves a naive method, and its precision is not very high. Improving upon this is our future task.

2.3 Design of atomic actions

Atomic actions are designed at the level of, for example, “proceed while avoiding obstacles”, and implemented on the Python side. Plans using these actions as the smallest units of action are prepared on the Jason side. As a result, when writing plans for the BDI agent, we can plan using atomic actions, which are robust against obstacles.

We implemented the following atomic actions. The heading of each entry shows the name of the action. Actions `forward_Qbo` and `search_Qbo` were implemented in our previous research[7, 8].

`looking_Qbo`: search for an object and take an argument as an ID of the target object.

First, obtain 3 images from the camera by rotating the robot’s head in different directions (covering about a 160° view) and splitting each image into 3 × 2 areas. Next, estimate whether the target object exists in each area using SVM. If, for some areas, the target is judged to exist at a certainty higher than some threshold, the target is considered to exist in the direction of the area with the highest certainty. If not, the target is considered not to exist. The result is returned as a perception.

`forward_Qbo`: take an argument as a base direction, turn to the specified base direction, and move ahead a constant length. The capability to detect an obstacle with ultrasonic sensors in parallel with moving is built into this action, and for detecting an obstacle, the robot stops. After this action, information of the robot’s direction and whether the

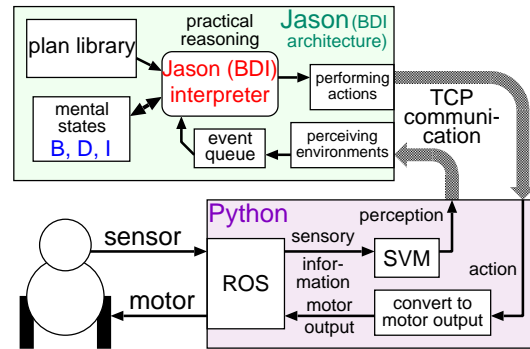


Figure 3: BDI agent and implementation of robot’s behavior

robot is facing an obstacle is received as perceptions (the same holds for `search_Qbo`).

`search_Qbo`: take an argument as a base direction, turn to the direction that is close to the specified base direction, and move to where no obstacles are found. This first involves turning to the specified direction, and if there is any obstacle in that direction, turning around little by little until facing no obstacle.

2.4 Decision-making program for robots

We prepared a decision-making program for our robot as a plan of the BDI agent as follows. Currently, we assume that a robot is in a room whose door is opened inward. The final goal for the robot is “to move out of the room”, and we set a goal of “reach the exit of the room” as an intermediate step. The plan to achieve this goal is written on Jason (as shown in Fig. 4) and consists of the following sequence of sub-goals: i) access the door of the room and ii) at that point, find the entrance of the room.

```
@re
+!reach_exit
<- // access the door
    !reach_object(door);
    // find the entrance
    !reach_object(entry).
```

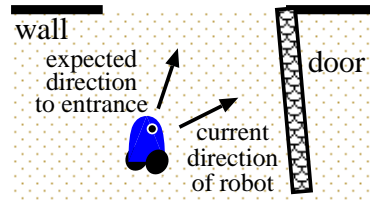


Figure 4: Plan for finding entrance of room Figure 5: Belief regarding direction to door and entrance

To achieve these sub-goals, we prepared sub-plan “ro” shown in Fig. 6 on Jason. Note that a tag beginning with “@” at the top of each plan shows the name of that plan. This plan, working with some sub-subplans, is prepared for reaching the front of a target object. It is commonly used in both i) and ii) above. In Fig. 6 (and for other figures referred in this section), a plan is in the form of “*triggering_event* : *precondition* <- *plan.body*” (where *precondition* can be omitted), and “!”, “+!” denote a sub-goal and event of goal addition, respectively. Note that not all definitions of sub-goals are given in this paper. The symbol “-+” denotes a belief revision. Atoms that appear in *plan.body*, which do not begin with any special symbol such as “!” or “-+”, are atomic actions; inside Jason, atomic actions are implemented using Java. The symbol “//” denotes a comment. Sub-plan “ro” executes the following processes.

- (1) Initially set the approximate direction of the target.
- (2) Receive perception.
- (3) Attempt to find the target object using `looking_Qbo`.
- (4) Depending on the perception caused by (3), decide the next action (`forward_Qbo` or `search_Qbo`) as follows (see Fig. 7), and send it to the Python side via TCP (by `tcp_write`).
 - ⟨a⟩ As a base direction, choose the direction toward the target if it is visible. If not, choose the direction initially given as an approximate direction to it.

```

@ro
+!reach_object(Object)
<- // initially set approximate direction to object
    !set_expected_target_direction(Object);
    !search_object(Object). // main process of search

@s_td1
+!set_expected_target_direction(door)
<- // set expected direction of door given ad hoc
    -+expected_target_direction(0).

@s_td2
+!set_expected_target_direction(entry)
:   my_current_direction(Dir)
<- // expected direction of door entry is separated
    // by 45 degrees from current direction
    -+expected_target_direction(Dir + 45).

@so1
+!search_object(Object)
:   not front_of(Object) // if not in front of object yet
<- // receive perception caused by previous action
    !get_my_current_direction;
    !get_information_of_obstacle;
    // internally call looking_Qbo to find target object
    !around_search(Object);
    // decide next action and send it to robot
    !decide_action;
    // recursively try to achieve the goal
    !search_object(Object).

@so2
+!search_object(Object)
<- true. // if already in front of object

```

Figure 6: Sub-plans for finding object

- (b) Select forward_Qbo to proceed if the robot is facing no obstacle, or search_Qbo to avoid the obstacle as the next action. The base direction chosen in (4)(a) is given as the argument of the action.

- (5) Repeat the process from (2) to (4) by recursive call.

Between i) and ii) mentioned above, to obtain the initial direction of the entrance, the robot uses an empirical rule that the expected direction to the room entrance is separated by 45° from the direction of the robot when it reaches the front of the door (plan “s_td2” in Fig. 6). This rule comes from the fact that the angle between the entrance and door was about 90° in this experiment (Fig. 5).

2.5 Discussion on process of recognizing door

In this study, the robot recognized the door via its visual perception. However, due to the limit of its range of vision, the robot could only acquire a partial image of the door instead of that of the entire door. In this study, the room door had a horizontal slit under the doorknob, which the robot selected as a characteristic of the door in order to recognize it. We argue

```

@da1
+!decide_action
: // if target is visible, and facing obstacle
  found_target(Object) & target_direction(Dir) &
  found_obstacle
<- // find nearest direction with no obstacle
  tcp_write(search_Qbo(Dir)).

@da2
+!decide_action
: // if target is visible, and not facing obstacle
  found_target(Object) & target_direction(Dir)
<- // proceed toward the target
  tcp_write(forward_Qbo(Dir)).

@da3
+!decide_action
: // if target is not visible, and facing obstacle
  found_obstacle & expected_target_direction(Dir)
  // expected_target_direction/1 returns approximate
  // direction of the target given initially
<- // find direction with no obstacle near initial direction
  tcp_write(search_Qbo(Dir)).

@da4
+!decide_action
: // if target is not visible, nor facing obstacle
  expected_target_direction(Dir)
<- // proceed toward initial direction
  tcp_write(forward_Qbo(Dir)).

```

Figure 7: Sub-plans for deciding action

that the procedure described above is close to the concentration of mind in human object recognition. When a human opens a door, he/she concentrates his/her consciousness on a characteristic part of the door, such as the doorknob, to recognize the door and open it, instead of having awareness of the entire door. We assume that localized use of information is also useful for robots to recognize their environments.

3 Experimental results

We now discuss the results of our experiment.

First, the robot moved in the expected direction of the door given initially, found the door, and moved in the direction toward the front of the door. When it approached the front of the door, the context of plan “so1” in Fig. 6 for reaching the front of the door was not satisfied under the following two conditions: recognizing the door by using the SVM and meeting the requirement of the ultrasonic sensors on its body, where we call these conditions “stop conditions” for plan “re” in Fig. 4. Then goal !reach object (door) (in plan “re”) was achieved through plan “so2” in Fig. 6. Therefore, the robot’s next goal was !reach_object(entry) (in plan “re”) for reaching the entrance and exiting the room, and the robot moved in the direction by setting the rule of sub-plan “s_td2” in Fig. 6 to find the entrance. With this knowledge, the robot moved to the entrance of the room and exited the room. This is a desirable result (Fig. 8).

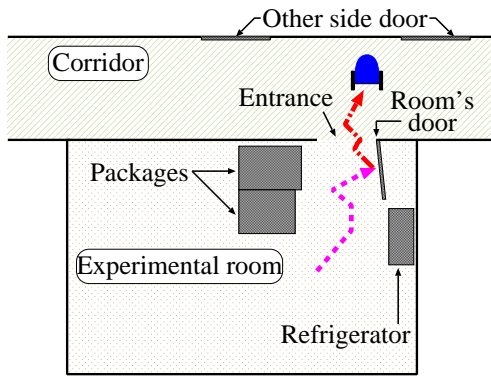


Figure 8: Route of robot's movement during experiment (dotted line denotes executing plan for reaching front of door. One-dot chain line denotes plan for reaching entrance of room.)

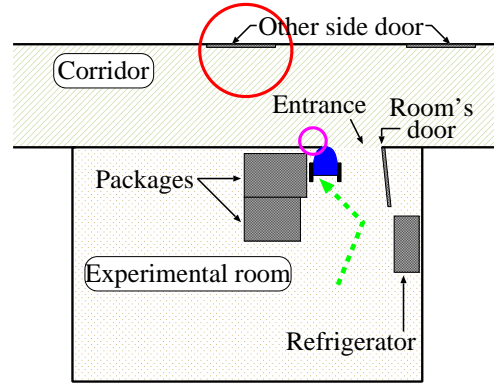


Figure 9: Route of robot's movement during experiment (dotted line is executing plan for reaching front of door. Large circle is condition of recognizing door. Small circle is condition of meeting requirement of ultrasonic sensors.)

Second, though the robot moved in the expected direction toward the door given initially, found the experimental room door, and moved in the direction toward the front of the door, like the first result, it found the opposite room door while it was moving toward the experimental room door. Therefore, the robot mistook the target as the opposite room door and moved toward the vicinity of the entrance of that room. At this point, the context of plan “so1” for reaching the front of the door was not satisfied because when the robot approached the entrance of the room, the stop conditions were incorrectly satisfied by finding the opposite door and meeting the requirement of the ultrasonic sensors since its body was caught on the edge of the doorway. Then goal `!reach_object(door)` (in plan “re”) was achieved with plan “so2”. Therefore, the robot started to exit at that location by executing goal `!reach_object(entry)` (in plan “re”). As a result, the robot could not find the entrance and exit the room by using this knowledge (Fig. 9).

4 Discussion

So far, we could not achieve our objective described in Sec. 1 through this experiment. This is mainly because the sense of robot's recognition is different from that of a human's. In the following, we list the important problems to be solved for improving the accuracy of the experiment.

4.1 Recognition of door

As the robot approached the door, the image of the door, which was small in the field of vision at first, gradually became larger. Then, as the robot neared the door, the entire door spread out from the view of the robot. As a result, the robot could not recognize that what was seen in the field of vision was the same object as the door that it had been looking at. This is because the robot uses only local visual information.

Furthermore, in our experiment, the robot could not find the difference between the nearby door and the opposite room door. The reason for this is considered to be that since

the robot classifies objects by feature extraction using the SVM based on camera images, it cannot distinguish objects with similar features from each other using perspective.

With these facts, the robot mistakenly recognized the door of the opposite room.

4.2 Stop conditions of plan in continuous space

From the second result in Sec. 3, the robot did not move properly due to improperly implementing the stop conditions. In this experiment, we implemented the rule for the following two stop conditions: “The front of the door” by recognizing the door, and meeting the requirement of the ultrasonic sensors. In the real world, we can understand “the front of the door” empirically, but it is difficult to implement such a condition. Because the real world is a continuous space, it is difficult to uniquely determine such an abstraction.

4.3 Method and amount of data used for learning

Currently, at most several hundred sets of image data are used for learning. This is considered to be a major factor of low precision. The accuracy of the experiment can be improved by increasing the amount of such data. However, it is a major problem that it takes considerable manpower and time to increase the amount of data.

There is also a possibility of improvement by introducing recent techniques, such as deep learning, for classifying door images.

As another problem, to recognize the door and approach it, it is necessary to keep recognizing that it is a door. Currently, the robot judges whether an image obtained from the camera is a door each time an action is taken. For this reason, if the recognition rate is not high, it tends to lose sight of the door. To address this problem, we should consider introducing methods such as object tracking.

4.4 Recognition of situation change

As a method of understanding the relative positional relation, we were thinking of using the rapid change of view when exiting a room. However, in our observation, the scenery observed by the robot changed much faster than with our eyes, even when it did not exit the room. This seems to be partly due to the low height of the robot.

4.5 Summary and future directions

We discussed problems to be solved to achieve our goal of a robot recognizing the change in the situation of exiting a room. To address issues and achieve our goal, one possible idea we are considering is to understand the shape of a building using 3D LIDAR (e.g. [9, 10]), and based on this information, make the robot be able to recognize that it has moved from a room to a corridor. One of our future tasks is to solve these problems and have the robot acquire the ability to obtain the logical relationship between the locations as knowledge and create a map by moving to the destination while achieving its subgoal.

5 Related work

Zhang et al. [11] proposed an autonomous mobility control method by combining high-level planning implemented in answer set programming (ASP) and low-level planning implemented in the partial observation Markov decision process (POMDP). By making high-

level inference using ASP, simulation results improved compared to behavior control only with POMDP. They also succeeded in experiments with real robots. While their method is based on self-location estimation using existing SLAM technology, we discussed a behavior control method without using SLAM.

Nüchter et al. [12] proposed a method for understanding the space of the real world by labeling floors, ceilings, and walls using the inference rules of Prolog based on the vertical or parallel relationship on a 3D map obtained from a robot's 3D scan and using 6D SLAM[13]. However, they mentioned that using HTN planning[14] for determining a robot's movement using this logical information is a subject for future analysis. In contrast, we used the BDI model and logic programming for determining a robot's movement, and the robot performed goal-oriented actions.

Lidoris et al. [15] conducted the ACE project for developing a robot that can reach a destination without an existing map or GPS sensors and move outdoors. The main contribution of their paper was that the robot could reach a destination by asking pedestrians. Their robot is composed of a finite state machine consisting of three action-decision modules "Active, Inactive, and PriorityCheck". Unlike that, we used the BDI model for the robot's action decision, and the robot could flexibly modify its goal.

Brandstatter et al. [16] presented a self-position estimation method other than SLAM. They solved localization and navigation problems using the internal discrete topological representation of the environment called an internal map. The internal map consists of locations that are candidates for the self-position of the agent, regions which are sets of the locations, and adjacency relation between the regions. An agent can estimate the location based on the positional relationship and distance of the landmark seen from the current position. Location estimation can also be used for navigating to the destination. While they only conducted simulation, we gave an implementation based on logic programming.

6 Conclusion

We implemented a system that acquires logical recognition of positions in space and moves inside that space and conducted an experiment to clarify the current problems. We have to further consider the stop condition of the plan, which is one of the implementation problems. We believe that problems related to perspective and identity, recognition using local information, and learning volume and method can be improved by improving hardware performance and by changing the handling method. Furthermore, with regard to the robot's recognition of changes to the situation, such as recognition of "exiting the room", which was the goal of this research, we are considering creating knowledge unique to robots by using dedicated hardware such as 3D LIDAR or combining various sensors. Through further experiments based on the above process, we aim to solve these problems and create intelligence for robots that cooperate with humans, such as housekeeping robots.

References

- [1] M. Fujita, Y. Goto, N. Nide, K. Satoh, and H. Hosobe. Autonomous control of mobile robots using logical representation of map and inference of location. In *Proc. of IEEE ICA2016*, pages 78–81, 2016.

- [2] A. S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In M. N. Huhns and M. P. Singh, editors, *Readings in Agents*, pages 317–328. Morgan Kaufmann, San Francisco, 1997.
- [3] M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, 1987.
- [4] R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.
- [5] M. Fujita, H. Katayama, N. Nide, and S. Takata. BDI robots who adapt to the diversity of the real world. *IPSJ Transactions on MPS*, 5(1):50–64, 2012. (In Japanese).
- [6] E. Fernández, L. S. Crespo, A. Mahtani, and A. Martinez. *Learning ROS for Robotics Programming — second edition*. Packt Publishing, 2015.
- [7] M. Fujita, Y. Goto, N. Nide, K. Satoh, and H. Hosobe. Logic-based and robust decision making for robots in real world. In *Proc. of AAMAS '14*, pages 1685–1686, 2014.
- [8] M. Fujita, Y. Goto, N. Nide, K. Satoh, and H. Hosobe. An architecture for autonomously controlling robot with embodiment in real world. In *Proc. of Knowledge Representation and Reasoning in Robotics (workshop at ICLP 2013)*, pages 59–71, 2013.
- [9] K. Kimoto, N. Asada, T. Mori, Y. Hara, A. Ohya, and S. Yuta. Development of small size 3D LIDAR. In *Proc. of IEEE ICRA2014*, pages 4620–4626, 2014.
- [10] Hokuyo automatic co., ltd. Scanning rangefinder distance data output/YVT-35LX product details. <https://www.hokuyo-aut.co.jp/search/single.php?serial=165>. Viewed on Dec 09, 2017 (In Japanese).
- [11] S. Zhang, M. Sridharan, M. Gelfond, and J. Wyatt. An architecture for knowledge representation and reasoning in robotics. In *Proc. of 15th International Workshop on Non-Monotonic Reasoning*, pages 233–241, 2014.
- [12] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.
- [13] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM — 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722, 2007.
- [14] M. Ghallab, D. S. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann Publishers Inc., 2004.
- [15] G. Lidoris, F. Rohrmüller, D. Wollherr, and M. Buss. The autonomous city explorer (ACE) project — mobile robot navigation in highly populated urban environments. In *Proc. of IEEE ICRA2009*, pages 1416–1422, 2009.
- [16] C. Brandstatter, S. Schaat, A. Wendt, and M. Fittner. How agents use breadcrumbs to find their way. *Journal of Computers*, 12(1):89–96, 2017.