Proc. 2005 IEEE Int. Conf. on Robotics and Automation,
pp. 3389-3394, Brcelona, Spain, April 2005.

Interactive Teaching of a Mobile Robot

Jun Miura, Koji Iwase, and Yoshiaki Shirai
Dept. of Computer-Controlled Mechanical Systems,
Osaka University, Suita, Osaka 565-0871, Japan
jun@mech.eng.osaka-u.ac.jp

***Abstract*— Personal service robots are expected to help people in their everyday life in the near future. Such robots must be able to not only move around but also perform various operations such as carrying a user-specified object or turning a TV on. Robots working in houses and offices have to deal with a vast variety of environments and operations. Since it is almost impossible to give the robots complete knowledge in advance, on-site robot teaching will be important. We are developing a novel teaching framework called *task model-based interactive teaching*. A task model describes *what knowledge is necessary* for achieving a task. A robot examines the task model to determine missing pieces of knowledge, and asks the user to teach them. By leading the interaction with the user in this way, the user can teach important (focal) point easily and efficiently. This paper deals with a task of moving to a destination at a different floor; the task includes not only the movement but also the operation of recognizing and pushing elevator buttons. Experimental results show the feasibility of the proposed teaching framework.**

I. Introduction

Personal service robots are one of the promising areas to which the robotics technologies can be applied. As we are facing the "aging society", the need for robots which can help human in various everyday situations is increasing. Possible tasks of such robots are: bringing a specified object to the user in the bed, cleaning a room, mobile aid, and so on. Recently several projects on personal service robots are going on [1], [2], [6], [22].

Service robots need to move in their working areas autonomously from one place to another, without collision. In addition, robots must be able to perform various operations such as carrying a user-specified object and turning a TV on. Robots working in houses and offices have to deal with a vast variety of environments and operations. Since it is almost impossible to give the robots complete knowledge in advance, on-site robot teaching will be important.

Approaches to robot teaching can roughly be divided into two categories. One is the *direct method* in which a user teaches a robot a trajectory, and the robot plays back the trajectory, as in the case of most industrial robots. The other approach is the *indirect method* in which a user teaches "what the task is" to the robot *indirectly*. "Teaching by demonstration" methods [4], [9], [15] are typical examples. The direct methods are easily implemented on robots but require much user's efforts, while the indirect methods are intuitive for users but may require a high-level ability of a robot such as recognizing human demonstrations and planning feasible motions.

We, therefore, takes an approach which exists in between these two; a user teaches several important (focal) points on a task in response to the queries from a robot. This approach does not require much effort of the user, and does not entail a very high recognition and inference ability of the robot. The robot is given in advance "what knowledge is necessary" for each task, in the form of *task models*. If a part of knowledge in the task model is initially missing, it will be acquired on-site from the user through interaction.

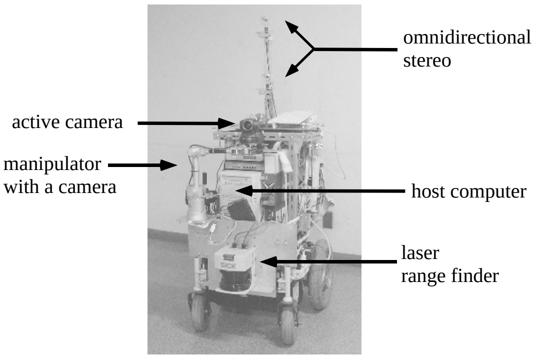


Fig. 1. Our mobile robot.

Several interactive teaching frameworks have been proposed such as acquiring missing heuristics for motion control by identifying them using a probabilistic knowledge model and by generating the corresponding queries [10], acquiring appropriate parameter values for predetermined sensing operations using GUI [20], and incremental robot programming using a multi-modal interface [8]. This paper aims to develop a general model of interactive teaching for complex robot tasks including sensing operations.

In this paper, we teach our mobile robot to take elevators (the *take-an-elevator* task); this task includes several important operations which usual service robots may perform, such as pushing an elevator's button, although the task of moving to another floor using an elevator itself may not be a service task. Fig. 1 shows our robot, which is a modified electric wheelchair controlled by a host computer. The robot has an omnidirectional stereo system [13], an active camera, a SICK laser range finder (LRF), and a four DOF arm with a camera. We will describe an implementation of teaching system and show several teaching examples and experimental results.

II. A Framework for Interactive Teaching

Interaction between the user and a robot is useful for an efficient and easy teaching of task knowledge. Without interaction, the user has to think by himself/herself about what to teach to the robot. This is difficult for the user partly because he/she does not have enough knowledge of the robot's ability (i.e., what the robot can (or cannot) do), and partly because the user's knowledge may not be well-structured. If the robot has knowledge of what is needed, then the robot can ask the user to teach it; this enables the user to easily give necessary knowledge to the robot.

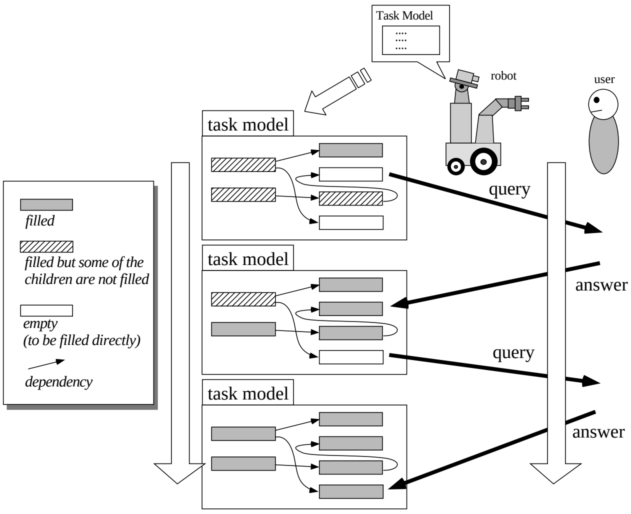


Fig. 2. Interaction process guided by task knowledge.

In our interactive teaching framework, the knowledge of a task is organized in the *task model*, in which necessary pieces of knowledge and their relationships are described. Some pieces of knowledge require other ones; for example, a procedure for detecting an object may need the shape or the color of the object. Such dependencies are represented by the network of knowledge pieces. Fig. 2 illustrates the interaction process guided by a task model. The robot determines missing pieces of knowledge (i.e., *empty* slots to be filled) in the task model, and generates queries for them. This examine-query-fill-in process leads the interaction with the user, and continues until the robot has the necessary and sufficient knowledge of the task.

III. Task Model and Interactive Teaching

This section explains the representations for task models and the teaching strategy.

A. Hierarchical Task Structure

Robotic tasks usually have hierarchical structures. Fig. 3 shows a hierarchy of robot motions for the *take-an-elevator* task. For example, a subtask, *move and push button*, is further decomposed into two steps (see the bottom of the figure): moving to the position where the robot can push the button, and actually pushing the button by the manipulator using visual feedback. Such a hierarchical task structure is the most basic representation in the task model.

Non-terminal nodes in a hierarchical task structure are *macros*, which are further decomposed into more specific subtasks. Terminal nodes are *primitives*, the achievement of which requires actual robot motion and sensing operations. The representations for such operations are explained in the following subsections. What primitives are used depends on the ability of the robot. The detailed example will be shown later.

B. Description of Robot and Objects

The robot model describes knowledge of the robot system such as the size and the mechanism of components (e.g., a mobile base and an arm) and the function and the position of sensors (e.g., cameras and range finders).

Properties of objects are divided into changeable and unchangeable ones. Unchangeable properties include geometric ones such as size and shape, kinematic ones such as a door mechanism, and photometric ones related to visual recognition operations. Changeable properties include a pose, which defines an object local coordinate system, and states such as the door's state (open or close). Changeable properties are usually accompanied with sensing operations for determining them.

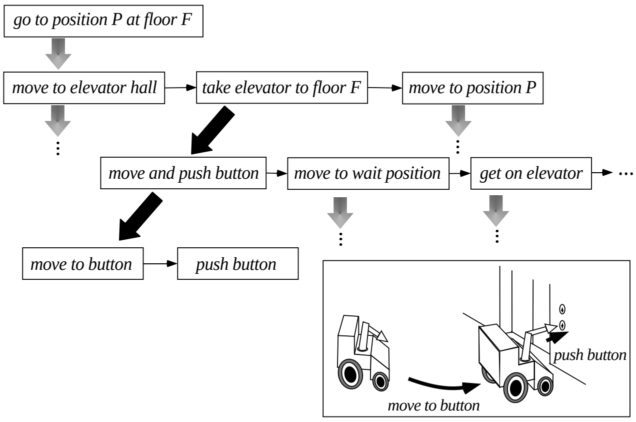


Fig. 3. A hierarchical structure of the *take-an-elevator* task.

C. Description of Sensor-Based Operations

1) Movements: The robot has two types of movements: free movement and guarded movement. A free movement is the one that the robot is required to reach a given destination without collision. The purpose of this movement is to reach the destination and the robot does not need to follow a specific trajectory. On the other hand, in a guarded movement, the robot has to follow some trajectory, which is usually generated from the configuration of surrounding obstacles; movements of this type are basically used for reaching a specific pose (position and orientation) or for passing through a narrow space.

Each movement has the conditions, if necessary, which need to be satisfied at the start of the movement, during the movement, and at the end of the movement, respectively. Each condition is described by a geometric relationship between the robot and a part of surrounding objects and is usually accompanied with a sensing operation which verifies the condition.

2) Hand Motions: A hand motion is to make the robot hand move to a specific pose, usually with respect to an object, as in the case of guarded movements. It also has the three types of conditions with sensing skills are attached if necessary.

The above robot motions are implemented and used as primitives. The details of the primitives will be explained in Secs. V-B and VI-A.

D. Sensing Skills

A sensing operation is represented by a *sensing skill*. Sensing skills are used in various situations such as detecting and recognizing objects, measuring properties of objects, and verifying conditions on the geometric relationship between the robot and the objects.

E. Dependency between Pieces of Knowledge

There are dependency relationships between pieces of knowledge as mentioned above. The knowledge for a task is, therefore, represented as a network of pieces of knowledge in the task model. Fig. 4 shows a part of the network

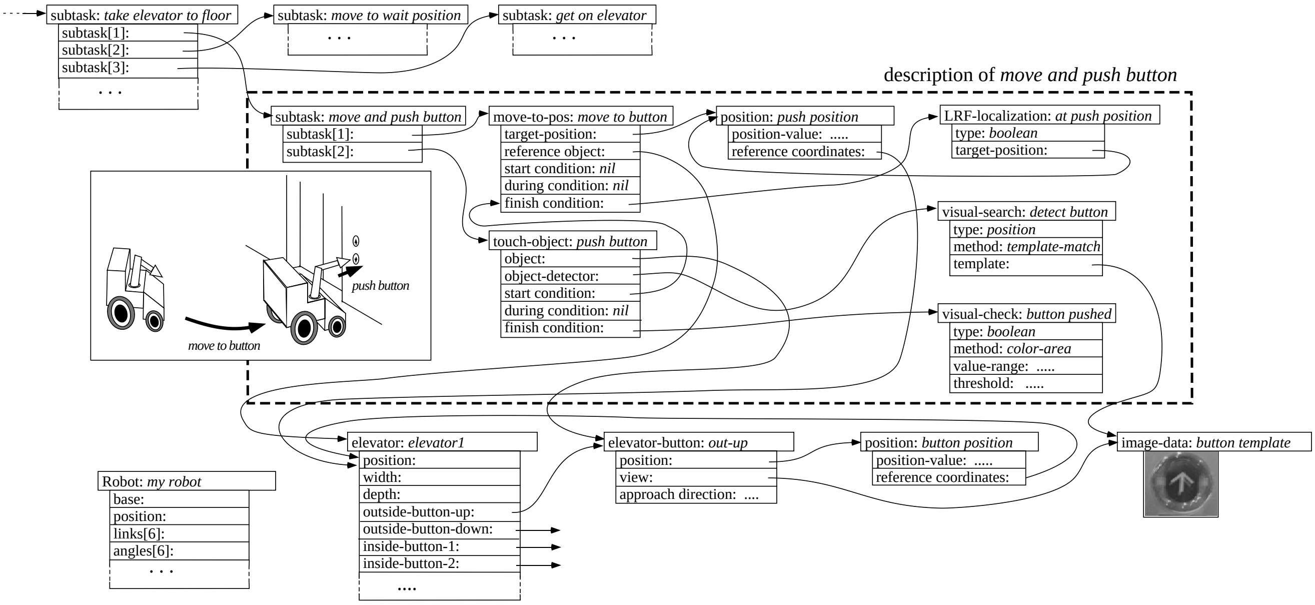


Fig. 4. A part of the network structure in the model for *take-an-elevator* task.

of the task model for the *take-an-elevator* task; in the figure, only the variables (*slots*) of each piece of knowledge and their dependency relationships are illustrated.

F. Interactive Teaching Using Task Model

The robot tries to perform subtasks in the task hierarchy one after another. When the robot cannot execute a motion because of a missing piece of knowledge, the robot pauses and generates a query to the user for obtaining it. By repeating this process, the robot completes the task model with leading the interaction with the user. It could be possible to examine the whole task model before execution and to generate a set of queries for a part of missing pieces of knowledge.

Each piece of knowledge needs the corresponding teaching method to be taught by the user. The level of information transferred through interaction obviously depends on the knowledge given to the robot, more specifically on the level of the primitives.

IV. Analysis of Take-an-Elevator Task

The approach we take in developing a robot system that can interactively learn from the user is as follows. We first analyze and the target task and develop the task model. We then pick up pieces of knowledge that cannot be instantiated off-site, and develop the corresponding teaching methods. This section describes the analysis result of the *take-an-elevator* task.

The task is decomposed into the following steps:

(1) Move to the elevator hall from the current position, provided that the route to the elevator hall is given.
(2) Move to the place in front of the button outside the elevator, where the manipulator can reach the button. The robot recognizes the elevator and localizes itself with respect to the elevator's local coordinates. For the movement, the robot sets a trajectory from the current position to the target position, and follows it by a sensory-feedback control.
(3) Localize the button and push it using the manipulator. The robot detects that the button is pushed by recognizing that the light of the button turns on.
(4) Move to the position in front of the elevator door where the robot waits for the door to open.
(5) Get on the elevator after recognizing the door's opening.
(6) Localize and push the button of the destination floor inside the elevator, as the same as (3).
(7) Get off the elevator after recognizing that the door opens (currently, the arrival at the target floor is not verified using floor signs inside the elevator car).
(8) Move to the destination position at the target destination floor, as the same as (1).

Based on this analysis, we developed the task model (a part of the model has been shown in Fig. 4).

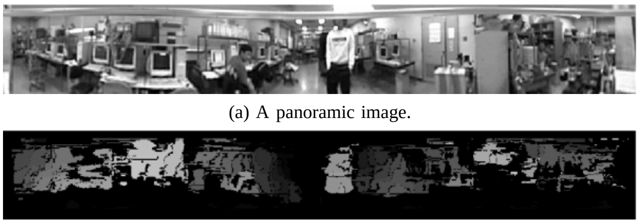
The above steps are divided into the following two parts. Steps (1) and (8) are composed of free movements. The other steps are composed of guarded movements near the elevator and hand motions. The following two sections explain the primitives and the teaching methods for the first and the second parts, respectively.

V. Map Generation, Motion Control, and Route Teaching

The robot needs a free space map and a destination (or a route to it) to perform a *free movement*. The *move-to-destination* primitive for free movements is realized by using the embedded functions of map generation and motion control. The destination or the route may be given by some coordinate values, but they are not intuitive for the user to teach. So we take the approach that we just take the robot to the destination and the robot memorizes the route to the destination.

A. Map Generation

Our map generation method uses both the omnidirectional stereo (see Fig. 5) and the LRF (laser range finder).



(a) A panoramic image.

(b) Panoramic disparity image obtained from (a).

Fig. 5. Omnidirectional stereo generates a panoramic disparity image.

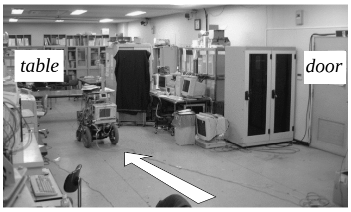
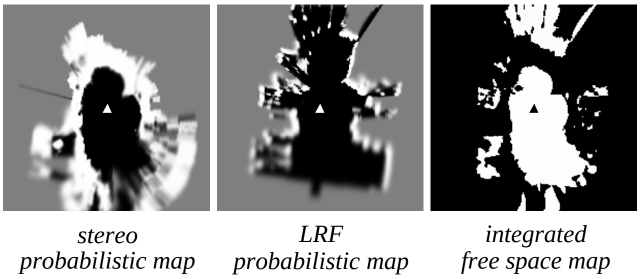


Fig. 6. An example scene.



stereo probabilistic map

LRF probabilistic map

integrated free space map

Fig. 7. Probabilistic maps and a free space map. Black and white triangles indicate the robot position and orientation [19].

By integrating data from both sensors, a reliable free space map is generated [19].

We keep a probabilistic occupancy map [5] for each sensor. Temporal integration of sensor data is carried out for each map separately using the probabilistic sensor uncertainty models. Ego-motion estimation, which is necessary for integrating sensor data obtained at different positions, is carried out by our scan matching-based estimation method [14] using both stereo and LRF data.

Then, the two probabilistic maps are integrated by first classifying each cell of a map into four classes (*free space*, *obstacle*, *undecided with observation*, and *undecided without observation*) and then integrating the classification results into the free space map [19].

Fig. 6 shows an example movement of our robot. Fig. 7 shows the maps generated after the movement. In the probabilistic maps, brightness indicates the probability of each cell being occupied by an obstacle. The maps are drawn in the robot coordinates.

B. Motion Control and move-to-destination *Primitive*

The robot generates a collision-free path given the current free space map and a goal position inside the free space. Considering the motion constraint of our robot driven by two powered wheels, we approximately represent a path of the robot by a sequence of circular segments. The robot can also adjust its speed by considering the map and motion uncertainty. Refer to [21] for the details of the motion planning and the speed selection. The method currently considers static environments.

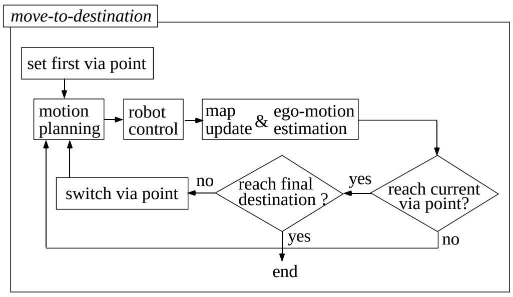


Fig. 8. *Move-to-destination* primitive.

Using the above map generation and motion control capabilities, we define the *move-to-destination* primitive for free movements. The process of the primitive is shown in Fig. 8; it controls the robot to pass near one via point to another until the robot gets sufficiently close to the final destination.

C. Route Teaching by Following Movement

We first take the robot to a destination. During this *guided* movement, the robot learns the route. Then the robot can reach the destination by localizing itself with respect to the learned route. Such two-phase methods have been developed for both indoor and outdoor mobile robots; some of them are map-based[12], [17] and some are view-based [11], [18].

In this work, the robot simply memorizes the trace of its guided movement obtained by our scan matching-based ego-motion estimation. Although the estimated trace may suffer from accumulated errors, the robot can safely follow the learned route because of the reliable map generation; the robot moves to the direction of the destination within the recognized free space.

The next problem is how to guide the robot. In [11], [12], we used a joystick to control the robot; but this requires the user to know the mechanism of the robot. A user-friendly way is to implement a person-following function to the robot [7], [24]. For a simple and reliable person detection, we use a teaching device which has red LEDs; the user shows the device to the robot while he/she guides it to the destination (see Fig. 9). The robot repeatedly detects the device in both of the two omnidirectional camera by using a simple color-based detection algorithm, and calculates its relative position in the robot coordinates. The calculated position is input to our path planning method [21] as a temporary destination. Fig. 10 shows a snapshot of person tracking during a guided movement.

From the trace of the robot positions during the guided movement, we set a sequence of via points, which the robot passes in the autonomous operation phase. Currently, via points are regularly placed on the trace with some interval (200 $[cm]$, at present).

Fig. 11 shows the given via points (“X” marks) and the trace of the autonomous movement from a starting position inside a room to the destination near the elevator. The white regions of the figure show the detected free space.

VI. Teaching of Vision-Based Operation

This section describes the methods for teaching the position of an elevator, the positions of buttons, and the views of them, after explaining the related primitives.

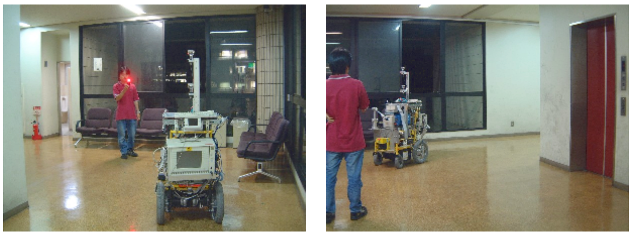


Fig. 9. Taking the robot to the destination.

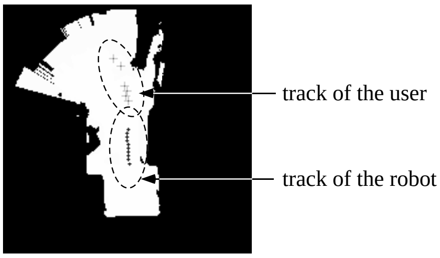


Fig. 10. Tracking the user. The white area is the detected free space.

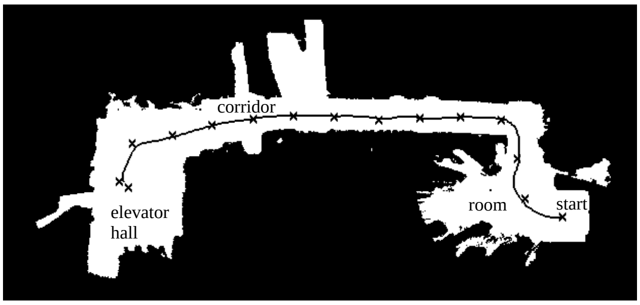


Fig. 11. A sequence of via points, estimated trace of the robot in the autonomous movement, and the recognized free space.

A. Primitives for Guarded Movement and Hand Motion

Move-to-pos primitive for a *guarded movement* requires a target pose of the robot and some reference object. This primitive repeatedly plans a trajectory and controls the robot to follow it until the robot comes sufficiently near the target pose.

Touch-object primitive for pushing an object requires the position of the object with the approaching direction. This primitive plans the hand trajectory and control the hand to follow it until some finish condition is satisfied.

B. Teaching the Elevator Position

Suppose that the robot has already be taken to the elevator hall, using the method described in the previous section. The robot then asks about the position of the elevator. The user indicates it by pointing the door of the elevator (see Fig. 12). The direction of the teaching device from the robot is detected by one of the omnidirectional cameras.

The robot has a general model of elevator shape, which is mainly composed of two parallel lines corresponding to the wall and the elevator door projected onto the floor. Using this model and the LRF (laser range finder) data, the robot searches the indicated area for the elevator and sets the origin of the elevator local coordinates at the center of the gap of the wall in front of the door (see Fig. 13). When getting on the elevator, the robot uses another model of the elevator car shape (i.e., a rectangular).

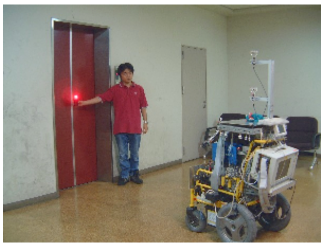


Fig. 12. Teaching the elevator position to the robot.

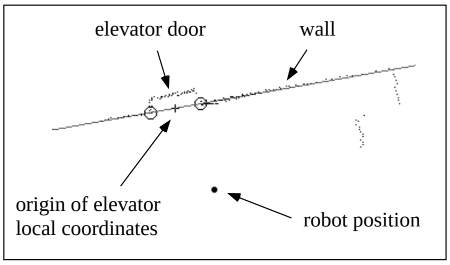


Fig. 13. Elevator detection from the LRF data.

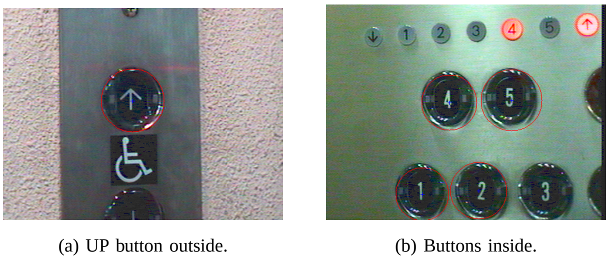

(a) UP button outside.
(b) Buttons inside.

Fig. 14. Detected buttons.

C. Teaching the Button Position

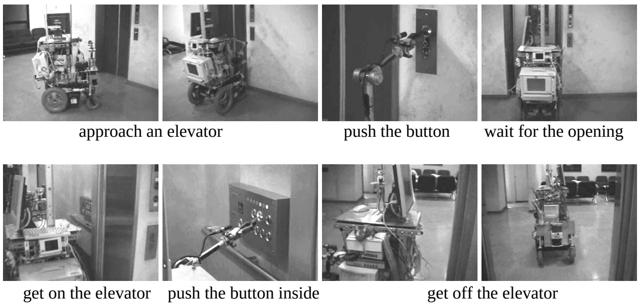
The robot then asks where the buttons are, and the user indicates their rough position. The robot searches the indicated area on the wall for image patterns which match the given button models (e.g., circular or rectangular) using a hough transform-based detection algorithm. Fig. 14(a) shows an example of detected button outside the elevator. The position of the button with respect to the elevator coordinates as well as the button view, which is used as an image template, is recorded after the verification by the user. The robot learns the buttons inside the elevator car in a similar way; the user indicates the position of the button box with respect to the elevator car coordinates. the robot searches there for buttons. Fig. 14(b) shows an example detection of buttons inside.

D. Autonomous Execution Result

The robot, having been taught the above-mentioned missing information from the user, succeeded in taking the elevator. Fig. 15 shows an execution of the task using the completed task model.

VII. Conclusions and Discussion

This paper has described a novel interactive teaching method, *the task model-based teaching*, for mobile robots. The task model describes what pieces of knowledge are



approach an elevator push the button wait for the opening

get on the elevator push the button inside get off the elevator

Fig. 15. The mobile robot is taking an elevator.

necessary for each task, including the following three kinds of robot-specific knowledge: object models, motion models, and sensing skills. Using the task model, the robot can determine what pieces of knowledge are further needed and plans necessary interactions with users to obtaining them. By this, the user was able to teach only the important pieces of task knowledge easily and efficiently. Preliminary implementations and experimental results have been shown on the *take-an-elevator* task.

Currently the task model is manually designed for the specific, *take-an-elevator* task from scratch, as a set of C++ methods. It would be desirable, however, that a part of existing task models can be reused for describing another. Since reusable parts are in general commonly-used, typical operations, a future work is to develop a repertoire of typical operations by, for example, using a generative approach [8], [16] or an inductive learning-based approach [3], [25]. By using the repertoire, the user's effort for task modeling is expected to be reduced drastically.

Another issue is the development of teaching methods. Although the mechanism of determining missing pieces of knowledge in a dependency network is general, for each missing piece, the corresponding method for obtaining it from the user should be provided. Such teaching methods are also designed manually at present and, therefore, the kinds of knowledge that can be taught is limited. Implementing the procedures for various knowledge requires much user's effort, especially for non-symbolic (e.g., geometric or photometric) knowledge. Another future work is thus to develop interfaces that can be used for teaching a variety of non-symbolic knowledge. Graphical user interfaces (GUIs) (e.g., [23]) or multi-modal interfaces (MMIs) (e.g., [8]) are suitable for this purpose.

Acknowledgments

This research is supported in part by Grant-in-Aid for Scientific Research from Ministry of Eduction, Culture, Sports, Science and Technology, the Kayamori Foundation of Informational Science Advancement, Nagoya, Japan, and the Artificial Intelligence Research Promotion Foundation, Nagoya, Japan.

REFERENCES

[1] MORPHA project, *http://www.morpha.de/*.

[2] R. Bischoff. HERMES – A Humanoid Mobile Manipulator for Service Tasks. In *Proceedings of 1997 Int. Conf. on Field and Service Robots*, pp. 508–515, 1997.

[3] B. Dufay and J.C. Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning. *Int. J. of Robotics Research*, Vol. 3, No. 4, pp. 3–20, 1984.

[4] M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann. Teaching Service Robots Complex Tasks: Programming by Demonstration for Workshop and Household Environments. In *Proceedings of 2001 Int. Conf on Field and Service Robots*, pp. 397–402, 2001.

[5] A. Elfes. Sonar-Based Real-World Mapping and Navigation. *Int. J. of Robotics and Automat.*, Vol. 3, No. 3, pp. 249–265, 1987.

[6] B. Graf, M. Hans, and R.D. Schraft. Mobile Robot Assistants. *IEEE Robotics and Automation Magazine*, Vol. 11, No. 2, pp. 67–77, 2004.

[7] E. Huber and D. Kortenkamp. Using Stereo Vision to Pursue Moving Agents with a Mobile Robot. In *Proceedings of 1995 IEEE Int. Conf. on Robotics and Automation*, pp. 2340–2346, 1995.

[8] S. Iba, C.J. Paredis, and P.K. Khosla. Interactive Multi-Modal Robot Programming. In *Proceedings of 2002 IEEE Int. Conf. on Robotics and Automation*, pp. 161–168, 2002.

[9] K. Ikeuchi and T. Suehiro. Toward an Assembly Plan from Observation Part I: Task Recognition With Polyhedral Objects. *IEEE Trans. on Robotics and Automat.*, Vol. 10, No. 3, pp. 368–385, 1994.

[10] T. Inamura, M. Inaba, and H. Inoue. Integration Model of Learning Mechanism and Dialog Strategy Based on Stochastic Experience Representation Using Bayesian Network. In *Proceedings of 2000 Int. Workshop on Robot and Human Interactive Communication*, pp. 247–252, 2000.

[11] H. Katsura, J. Miura, M. Hild, and Y. Shirai. A View-Based Outdoor Navigation Using Object Recognition Robust to Changes of Weather and Seasons. In *Proceedings of 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2974–2979, 2003.

[12] K. Kidono, J. Miura, and Y. Shirai. Autonomous Visual Navigation of a Mobile Robot Using a Human-Guided Experience. *Robotics and Autonomous Systems*, Vol. 40, No. 2-3, pp. 121–130, 2002.

[13] H. Koyasu, J. Miura, and Y. Shirai. Realtime Omnidirectional Stereo for Obstacle Detection and Tracking in Dynamic Environments. In *Proceedings of the 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Sysetms*, pp. 31–36, 2001.

[14] H. Koyasu, J. Miura, and Y. Shirai. Integrating Multiple Scan Matching Results for Ego-Motion Estimation with Uncertainty. In *Proceedings of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3104–3109, 2004.

[15] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by Watching: Extracting Resuable Task Knowledge from Visual Observation of Human Performance. *IEEE Trans. on Robotics and Automat.*, Vol. 10, No. 6, pp. 799–822, 1994.

[16] S. Lauria, G. Bugmann, T. Kyriacou, and E. Klein. Mobile Robot Programming Using Natural Language. *Robotics and Autonomous Systems*, Vol. 38, No. 3/4, pp. 171–181, 2002.

[17] S. Maeyama, A. Ohya, and S. Yuta. Autonomous Mobile Robot System for Long Distance Outdoor Navigation in University Campus. *J. of Robotics and Mechatronics*, Vol. 9, No. 5, pp. 348–353, 1997.

[18] Y. Matsumoto, M. Inaba, and H. Inoue. Visual Navigation using View-Sequenced Route Representation. In *Proceedings of 1996 IEEE Int. Conf. on Robotics and Automation*, pp. 83–88, 1996.

[19] J. Miura, Y. Negishi, and Y. Shirai. Mobile Robot Map Generation by Integrating Omnidirectional Stereo and Laser Range Finder. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 250–255, 2002.

[20] M. Mizukawa, Y. Nakamura, H. Ogata, and Y. Yanagihira. Multi-Modal Information-Sharing Teaching System in Sensor-Based Robotics – Structured Description based on Task Properties that Enables Onsite-Tuning by Reflecting Skills. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1535–1540, 2002.

[21] Y. Negishi, J. Miura, and Y. Shirai. Adaptive Robot Speed Control by Considering Map and Localization Uncertainty. In *Proceedings of the 8th Int. Conf. on Intelligent Autonomous Systems*, pp. 873–880, 2004.

[22] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun. Towards Personal Service Robots for the Elderly. In *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000.

[23] F. Saito and T. Suehiro. Toward Telemanipulation via 2-D Interface – Concept and First Result of Titi. In *Proceedings of IECON 02*, pp. 2243–2248, 2002.

[24] Y. Sawano, J. Miura, and Y. Shirai. Man Chasing Robot by an Environment Recognition Using Stereo Vision. In *Proceedings of the 2000 Int. Conf. on Machine Automation*, pp. 389–394, 2000.

[25] M. Tsuda, H. Ogata, and Y. Nanjo. Programming Groups of Local Models from Human Demonstration to Create a Model for Robotic Assmebly. In *Proceedings of 1998 IEEE Int. Conf. on Robotics and Automation*, pp. 530–537, 1998.