IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 26, 2021, accepted October 8, 2021, date of publication October 14, 2021, date of current version October 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120216

ASTRON: Action-Based Spatio-Temporal Robot Navigation

YOSUKE KAWASAKI[1], (Member, IEEE), SHUNSUKE MOCHIZUKI[1], AND MASAKI TAKAHASHI[2], (Member, IEEE)

[1]School of Science for Open and Environmental Systems, Graduate School of Science and Technology, Keio University, Kohoku-ku, Yokohama 223-8522, Japan
[2]Department of System Design Engineering, Faculty of Science and Technology, Keio University, Kohoku-ku, Yokohama 223-8522, Japan

Corresponding author: Yosuke Kawasaki (y-kawasaki@keio.jp)

This work was supported by the Core Research for Evolutional Science and Technology (CREST) of the Japan Science and Technology Agency (JST) under Grant JPMJCR19A1.

ABSTRACT To achieve the tasks provided by a user, it is necessary for robots to have a plan that fully exploits their functionalities in an environment. The objective of this study is to realize robot task planning in real space for effectively use of the robot's functions The plan is formed by deriving a feasible action sequence by interpreting the instructions within the scope of the action possibilities of the robots and the changes in them. In this paper, we first propose an action graph as a novel environmental representation approach to facilitate the understanding of the robot's action possibility in real space. In the action graph, the action possibility is represented by nodes, which describe the spatial position to perform each feasible action, and edges, which describe the feasible actions, based on the subsystem-level affordance and the arrangement of objects in the environment. We also propose an action-based spatio-temporal robot navigation (ASTRON), which focuses on robot navigation tasks. ASTRON enables the robots to determine a feasible action sequence that utilizes their functions by interpreting the instructions based on the action graph. The effectiveness of the proposed method was evaluated through simulations and actual machine experiments in a coffee shop environment. In the actual machine experiments, the proposed method was applied to robots with different subsystem configurations. The experimental results demonstrated that the proposed method could plan the feasible action sequence to complete the tasks by considering the environmental state and the subsystem configurations of the robot.

INDEX TERMS Autonomous robots, task and motion planning, semantic scene understanding, action graph.

I. INTRODUCTION

With the diversification of robot types in recent years, the functions of each robot have also become more diverse. Therefore, to complete a task set by a user, a planning method that enables the robot to exploit most of its functions, such as mobile bases, manipulators, and speech mechanisms, is necessary. In conventional task planning, action sequences for achieving instructions are planned using task representations in symbolic space, such as STRIPS and PDDL [1]–[3]. However, task planning in symbolic space has the following two problems. First, it is difficult to consider the feasibility of the robot's actions in a real space as it depending on the positional relationship between the object set and the robot. Second, it is challenging to consider the changes in the environment in real space, such as changes in the arrangement of objects due to the execution of actions by the robot and changes in the feasibility of the robot's actions due to these environmental changes. Therefore, the sequence of actions planned in symbolic space may include actions that the robot cannot actually perform. In addition, the planning may not consider the actions that the robot can originally perform. In this study, we propose a method to realize task planning in real space. The requirements for task planning in real space are as follows. First, it is necessary to understand the feasible actions for the robot and the change in them due to the robot's actions, considering the functions that the robot has and the set of objects in the environment. Furthermore, to incorporate the understood feasible actions in real space into the planning problem, an environmental representation method that describes the connections of the possible actions is necessary.

The associate editor coordinating the review of this manuscript and approving it for publication was Seung-Hyun Kong.

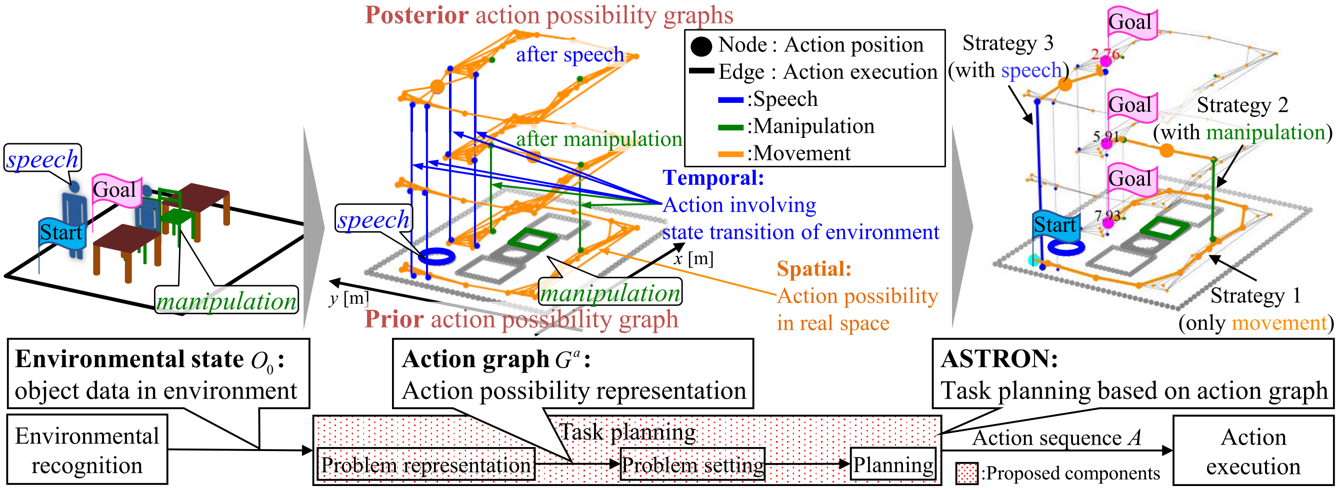


FIGURE 1. Overview of ASTRON.

A conventional method for representing the environment for task planning or motion planning is the semantic map, which describes the presence and the attributes of objects at each point in real space [4], [5]. In addition, a 3D scene graph describes not only the attributes of each object but also the positional relationships between the objects [6]. In the 3D scene graph, the environment structure is represented by a graph in which the nodes describe the object attributes, such as the position and the object label, and the edges describe the positional relationships between the objects. The purpose of these representations is to provide more accurate descriptions of the object types and the object positions in the environment. One approach that supplements the environment representation with action-related information is adopting the concept of affordance and adding information about action associated with the objects to the object information [7]. However, the feasibility of the robot's actions and the specific positions of action execution, which are necessary for task planning in real space, are not considered in this approach. Thus, an environmental representation method, which aims to understand the possible actions (that is, the actions that the robot can perform in real space and the execution position of the actions) and the changes in them caused by the action involving state transition of the environment, is necessary.

The objective of this study is to realize a task planning method in real space, which devises the executable action sequence by interpreting the instructions within the action possibilities of robots (spatial) and the changes in them (temporal). To achieve this objective, we propose the following two methods, as shown in the overview schematic in Fig. 1: i) First, an action graph is proposed as an environmental representation that describes the connection of feasible actions for the robot in real space. ii) As a design example of task planning in real space based on the action graph, an action-based spatio-temporal robot navigation (ASTRON) method is proposed, which focuses on robot navigation task. The center of Fig. 1 shows the action graph. The action graph is an environmental representation that describes the action possibility of the robot and the change in them by the connections between actions based on the environmental state, such as the properties and the arrangement of the objects recognized by the robot. The connections between feasible actions are represented by the action graph. In this graph, the nodes describe the specific positions at which the actions are to be performed, and the edges describe the feasible actions, such as movement (orange edges), speech (blue edges), and manipulation (green edges). To effectively use the robot's functions, the types of feasible robotic actions in real space and their execution positions are represented in a graph called the action possibility graph, based on the subsystem-level affordances that consider the subsystem configuration of the robot and the environmental state. Examples of speech actions with a robot's speech mechanism include asking a person to move from a location or to remove a movable object. In addition, an example of a manipulation action is removing a movable object from its location with the robot's manipulation mechanism. The specific execution positions of the feasible actions are determined by considering the position's reachability based on the arrangement of the object set and the distance of the robot from an object suitable for performing an action given as prior knowledge. The action graph is a multi-layered graph consisting of action possibility graphs before and after the state transition of the environment to describe changes in action possibilities due to actions involving the state transition, such as speech and manipulation. The actions involving the state transition of the environment are described by the edges which connect the action possibility graphs in the action graph (vertical blue or green edges). In action graph-based task planning, a feasible action sequence to accomplish a navigation task is planned within the actions feasible for the robot in the environment based on the action graph. Specifically, Dijkstra's method is applied to the constructed action graph, as shown on the right side of Fig. 1, to obtain the action sequence to attain the target position. In planning, the action sequence is determined after obtaining the action sequences following each strategy, that is, each possible state transition of the environment caused by the robot to achieve the instruction. In the example in Figure 1, three strategies are derived: strategy 1, which involves movement without any environmental change; strategy 2, which involves removing the chair; and strategy 3, which involves asking people. The candidates include strategies that may seem to go a long way around, at first sight, depending on the situation. Therefore, for example, action sequences in which a robot, instead of approaching the goal directly, approaches a person in a direction different from the goal, and requests for help in reaching the goal, can be performed.

This study makes the following contributions: i) Understanding the connections of feasible actions in the environment is realized with the action graph, where the nodes describe the positions at which the robot can perform each feasible action, and the edges describe the feasible actions based on the subsystem-level affordance and the arrangement of objects in the environment. This enables the planning of the feasible action sequences which makes the robots effective use of their capabilities. ii) The simultaneous representation of the action possibility and the changes in it caused by the robot action is realized using an action graph that has a multi-layer construction. This allows the representation of various actions in the action graph and the understanding of the action sequences following candidate strategies to achieve the instructions. The effectiveness of the proposed method was evaluated through simulations and actual machine experiments in a coffee shop environment.

II. RELATED WORK

A. TASK PLANNING AND MOTION PLANNING

Our work is related to task planning and motion planning methods that consider the feasibility of the action in real space. One of the conventional approaches is the task and motion planning (TAMP) method which combines task planning in symbolic space and motion planning in the configuration space [8]–[10]. The other approach is navigation among movable obstacles (NAMO), which is a navigation method that considers not only avoiding obstacles but also interaction with movable obstacles [11], [12]. The feasibility of actions in real space is verified using these methods based on the following two procedures. First, the path to the goal of each action is calculated for an environment, excluding the movable objects through a motion planning method. Then, the intersections of the calculated path with the excluded movable objects are verified. If they are intersected, it is difficult to achieve the verified action, and the interference with the intersecting object is the required action (requirements) for the verified action. In the motion planning method, the path is calculated to directly lead to the goal. Therefore, it is difficult to determine the action sequence that may be seen to go a long way around at first sight, with an approach to realize the action feasibility and the requirements for the action based on motion planning.

In contrast, in the proposed method, the action possibilities of the robot in real space can be understood by constructing an action graph based on the set of objects identified by the robot and the robot's subsystem configuration. In addition, in the task planning based on the constructed action graph, it is possible to plan an action sequence for each selectable state transition of the environment to reach the goal. This enables the robot to grasp selectable strategies in the environment, including those that do not directly approach the goal and to select an appropriate strategy.

B. ENVIRONMENTAL REPRESENTATION METHOD

Our work is also related to the methods of environmental representation. One of the classic environmental representations is the occupancy map obtained using the simultaneous localization and mapping (SLAM) method [13]. The occupancy map describes the presence of objects at each point in real space. The environmental map, which describes object attributes such as object labels as well as the presence of objects, is called the semantic map [4], [5]. Additionally, a 3D scene graph, which describes not only the attributes of each object but also the positional relationship between the objects, was proposed [6]. These environmental maps are intended to provide more accurate descriptions of objects in the environment. The data from the environmental representations have been applied to traditional task planning methods in symbolic space.

Environmental representations that describe both the data related to the object and to the action in the environment have been proposed. Action maps have been proposed as environment representations focusing on actions, which embed the action possibility in real space based on the history of human activities [14], [15]. In addition, another approach is to apply an object classification method based on the concept of affordance to associate the actions with objects [16], [17].

Additionally, to utilize environmental representations for motion planning methods in real space, representations, that describe not only information related to individual actions but also the connection of actions, particularly movement actions, are proposed. The occupancy map [13] and the topological map [18], [19] describe the connections between sources and destinations using a graph as possible movement actions in the environment based on the robot's traversability. In the multi-layer environmental affordance map, the connections between movement actions considering human activities are reflected in the occupancy map based on the traversability and the concept of the space inferred from the object set [20].

In contrast, in the proposed action graph, it is possible to represent changes in the action possibility due to actions involving the state transition of the environment by adopting a multi-layered graph representation. This allows representing the connections between the multiple types of feasible actions, including movement and actions involving state transition of the environment, such as speech and manipulation.

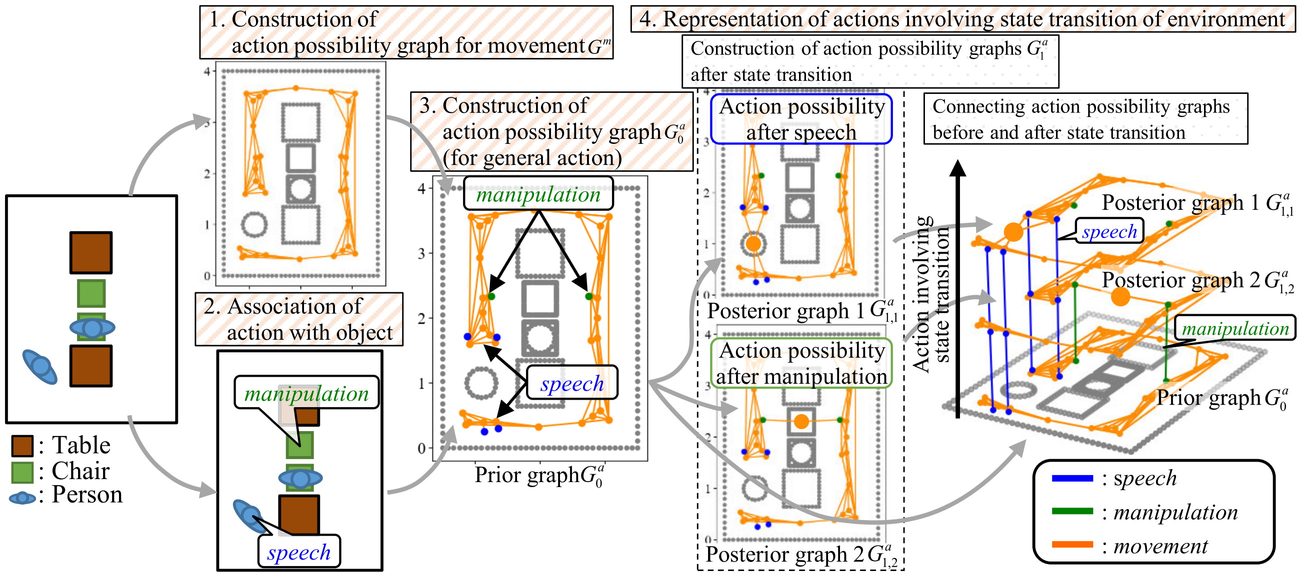
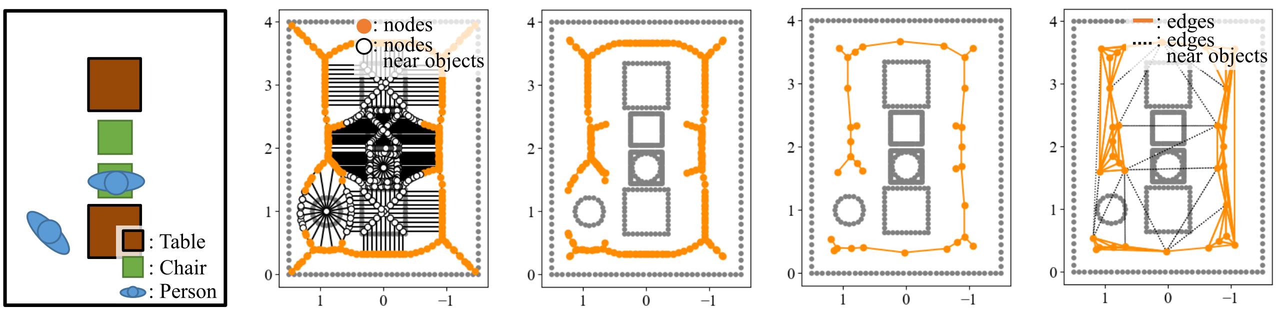


FIGURE 2. Action graph constructing pipeline.



(a) Example of environment (b) Generalized Voronoi diagram (c) Extracting valid nodes considering objects (d) Simplifying graph for efficient expression (e) Applying shortcut based on Delaunay triangulation

FIGURE 3. Construction of action possibility graph for movement.

III. ACTION GRAPH

A. OVERVIEW

The action graph G^a represents the action possibility (i.e. the actions that the robot can performs, the objects on which the actions are performed, and the position in real space where the actions are performed) and the changes in it by the connection of actions based on a recognized environmental state O_0.

The environmental state $O_0 = \{o_1, o_2, \ldots, o_{N^o}\}$ consists of a set of N^o objects. Examples of objects in this study are people, chairs, and tables. The object o has the following information:

- l^o: Label of the object
- $\boldsymbol{p}^o$: Position of the object in a two-dimensional absolute coordinate system
- θ^o: Direction of the object in a two-dimensional absolute coordinate system
- s^o: Shape data of the object consisting of geometric primitives such as square or circle and size. People are represented by circles, and chairs or tables are represented by squares.

The action graph G^a defines a pair of sets $G^a = (N, E)$ where N and E denote the set of nodes and edges, respectively. The nodes, n, represent the positions where the robot can perform actions in real space. The information contained in each of the nodes n is as follows.

- $\boldsymbol{p}^a$: Position at which to perform the action in a two-dimensional absolute coordinate system

In addition, actions a are assigned to the edges e. In other words, a transition between the nodes implies performing set action a. The information in each of the edges e is as follows.

- a: Action to be performed
- c^a: Cost of executing the action

Action a contains the action label, the target object data, the influenced object data, and other attributes necessary to perform the action. The specific attributes of action a are described in section III-C.

The action graph G^a consists of multiple action possibility graphs. The action possibility graph describes feasible actions for the robot and the position at which the actions are performed based on the environmental state and the subsystem configuration of the robot. The positions at which the feasible actions are performed are represented by the nodes of the action possibility graph. The edges of the action possibility graph represent feasible movement actions in which the robot moves in real space. In addition, the feasible action involving the state transition of the environment is represented by the edges connecting a prior action possibility graph G_0^a with a posterior one G_1^a to the state transition because the action possibility is changed owing to the state transition. By constructing an action graph with these multiple action possibility graphs, it is possible to clearly describe the action to be performed and the change in the corresponding action possibility. If the prior action possibility graph to the state transition of the environment G_0^a does not contain any actions involving the state transition of the environment, the action graph G^a is equal to the action possibility graph G_0^a, as follows:

$$G_0^a = G^a. \tag{1}$$

However, if the prior action possibility graph G_0^a contains actions involving the state transition of the environment, the action graph G^a contains the prior action possibility graph G_0^a and the posterior action possibility graphs $G^a_{1,1\sim N_1^O}$ to the state transition.

$$G_0^a, G_{1,1}^a, \ldots, G^a_{1,N_1^O} \in G^a, \tag{2}$$

where N_1^O is the number of possible environment states caused by the actions of the robot from the prior environment state O_0.

The design concept for the automatic construction of the action graph is shown in Fig. 2.

1) Constructing the action possibility graph G^m for understanding where the robot can reach in the environment (section III-B)
2) Associating actions with objects in the environment based on the subsystem-based affordance (section III-C)
3) Constructing action possibility graph G_0^a to represent the possibility of actions other than movement (section III-D)
    a) Selecting positions to perform the actions associated with the objects among from the robot's reachable positions (section III-D1)
    b) Adding the nodes and edges related to the associated actions to the graph G^m based on the action property and the selected reachable positions (section III-D2)
4) Representing actions involving state transition of environment (section III-E)
    a) Constructing posterior action possibility graphs to actions involving the state transition of the environment $G^a_{1,1\sim N_1^O}$ (section III-E2)
    b) Constructing action graph G^a by connecting the prior action graph G_0^a and the prior action graphs $G^a_{1,1\sim N_1^O}$ (section III-E3)

The details of these processes are described below.

B. CONSTRUCTION OF ACTION POSSIBILITY GRAPH FOR MOVEMENT

To sparsely represent where the robot can reach, the action possibility graph for movement G^m is constructed by considering the arrangement and size of the recognized objects in the environment. To understand the movement actions feasible for the robot, we constructed this graph in a two-dimensional space. The graph construction process is detailed below using the example environment shown in Fig. 3a.

1) GENERATION OF GENERALIZED VORONOI DIAGRAM

First, the footprint of the objects in the environment, including the wall, is expressed as a set of points in a two-dimensional space. Then, the generalized Voronoi diagram is generated by setting the base points of the diagram equal to the points of the object, as shown in Fig. 3b.

2) EXTRACTING VALID NODES CONSIDERING OBJECTS

Because the generalized Voronoi diagram contains untraversable edges and nodes unreachable by the robot, the invalid nodes and edges are deleted by considering the robot's footprint, as shown in Fig. 3c.

3) SIMPLIFYING GRAPH FOR EFFICIENT EXPRESSION

The graph is simplified to express where the robot can reach efficiently, as shown in Fig. 3d. The two connected nodes, which have only two edges, are integrated into one node. Then, the integration is adopted only if the new edge is not close to objects. By performing this integration process repeatedly, the number of nodes is reduced.

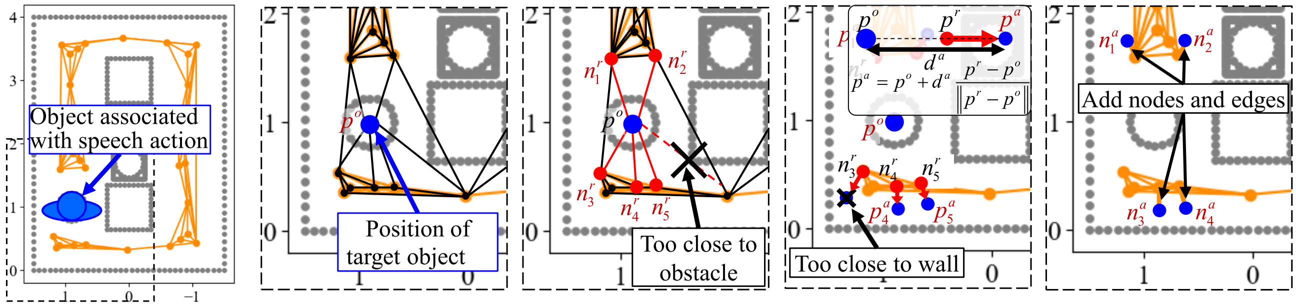
4) APPLYING SHORTCUT BASED ON DELAUNAY TRIANGULATION

To improve the navigation efficiency, the connectable nodes to each other are connected by the edge. The connectivity is checked based on the intersection of the objects with the edge candidates obtained by applying Delaunay triangulation to the nodes.

Finally, the obtained action possibility graph for movement G^m is as shown in Fig. 3e. Because the edges of the action possibility graph for movement G^m represent the *movement* action, *movement* is assigned to the action label l^a, and the costs of actions c^a are calculated based on the evaluation index of the task planning and the length of the edges.

C. ASSOCIATION OF ACTION WITH OBJECT

The action provided between the robot and the objects depends on the configuration of the robot subsystem. In addition, the action depends on not only the properties of each object but also the objects around it. Therefore, the actions are associated with the object set based on the subsystem-level affordance, which is defined as the affordance considering the robot's subsystem configuration. In this study, we assume that robots can acquire prior knowledge about the affordances and actions, as shown in Tables 1 and 2 with the development



(a) Action possibility graph for movement G^m (b) Delaunay triangulation (c) Selecting reachable nodes n^r (d) Determining action position p^a (e) Action possibility graph

FIGURE 4. **Construction of action possibility graph.**

TABLE 1. **Prior knowledge of subsystem-level affordance.**

	object set	Person	Chair	Table	Person, Chair	Chair, Person
subsystem	mobile base	none	none	none	none	none
	speaker	*speech*1	none	none	*speech*2	none
	manipulator	none	*manipu−lation*	none	none	none

TABLE 2. **Prior knowledge of action.**

action label l^a	*movement*	*speech*1 "Let me through"	*speech*2 "Please remove chair"	*manipulation* remove object
target object o_t^a	none	person	person	chair
appropriate distance d^a from o_t^a	none	d^s e.g., 0.5 m	d^s e.g., 0.5 m	d^m e.g., 1.0 m
influenced object o_e^a	none	itself	chair	itself
effect e^a	none	*removed*	*removed*	*removed*

of the affordance classification methods [16], [17] and the action understanding methods [21], [22], Then, the actions are associated with the recognized object sets based on the prior knowledge and the robot's subsystem configuration.

The action labels l^a are associated with the recognized object sets, considering the combination of an object set and a subsystem of the robot based on Table 1. In this study, the assumed types of subsystems are mobile base, speaker, and manipulator. Examples of action labels l^a include *movement*, *speech*1 in which the robot asks a person to let it to move through, *speech*2 in which the robot asks a person to remove a chair, and *manipulation* in which the robot removes a chair. The details of the subsystem-level affordance are described as follows: The *speech*1 action, which requires a speaker, is associated with a single person, and the *manipulation* action, which requires a manipulator, is associated with a single chair. Then, focusing on the object sets, no action label is associated with the person sitting on the chair, and the *speech*2 action, which requires a speaker, is associated with a person near a chair. Specifically, *speech*2 is assigned if the distance between a person and a chair is within a specific affordance range $r^{speech2}$,

The action properties corresponding to the action labels are shown in Table 2. The properties of the action correspond to action a embedded in the edges of the action graph G^a.

Each property of the action is described below.

- l^a: Label of the action
- o_t^a: Target object that the robot acts on
- d^a: Appropriate distance from the target object required to perform the action
  This is essential information for placing the symbolic actions in real space.
- o_e^a: Influenced object as a result of the robot's actions
  If the target object and the influenced object are the same, then the influenced object becomes *itself*. In addition, in the case of *speech*2, the influenced object becomes *chair*.
- e^a: Effect of the action on the influenced object
  In this study, an example of the effect is *removed*. *removed* implies that the influenced object will be removed, and its location will be movable.

The prior knowledge of affordances and actions shown in Tables 1 and 2 is a design example that can be extended.

D. CONSTRUCTION OF ACTION POSSIBILITY GRAPH

The execution positions of general action, including not only the *movement* action but also the actions associated with objects, are represented in an action possibility graph G_0^a. The requirements to ensure that a position is suitable for performing the action are the reachability to the position and from the position to the target object o_t^a based on the robot's footprint and the surrounding obstacles, and an appropriate distance to the target object o_t^a.

The action possibility graph G_0^a is constructed by adding nodes and edges related to every action associated with the objects to the action possibility graph for movement G^m based on the following procedure shown in Fig. 4.

1) SELECTING REACHABLE NODES

The reachable nodes n^r, whose position is reachable to the target object of action o_t^a, are selected from the nodes of G^m.

First, the Delaunay triangulation is applied to the points that include the nodes of G^m and the position corresponding to the target object p^o, as shown in Fig. 4b. Next, the approachable nodes n_r are selected, as shown in Fig. 4c. The candidates of n_r are the nodes that are connected to the position of the target object p^o in the result of the Delaunay triangulation. Then, n^r are adopted among the candidates by checking the reachability, whether the edges related to the candidate intersect with the surrounding obstacles considering the robot's footprint, as shown in Fig. 4c.

2) ADDITION OF NODES CORRESPONDING TO POSITIONS FEASIBLE FOR ACTION

To facilitate action execution, the positions to execute the action p^a are set based on the selected reachable nodes and the appropriate distance from the target object d^a, which is one of the action properties given as the prior knowledge. In addition, the action possibility graph G_0^a is constructed by adding nodes n^a and edges related to p^a to G^m.

First, the candidates for the position for the action p^a are set to the required distance d^a from the position of the target object p^o and, on a straight line connecting the target object p^o and the position of the selected reachable nodes p^r, as shown in Fig. 4d and follows:

$$p^a = p^o + d^a \frac{p^r - p^o}{\|p^r - p^o\|}, \tag{3}$$

Then, p^a are adopted among the candidates by checking the reachability, whether the robot can reach from the candidate's position to the target object p^o considering the surrounding obstacles and the robot's footprint, as shown in Fig. 4d. Finally, the action possibility graph G_0^a is constructed by adding n_a and the edges that connect n^a with n^r to G^m, as shown in Fig. 4e. The action labels l^a of all edges in the action possibility graph G_0^a are set to *movement*. The cost of the edges c^a is calculated based on the length of the edges and the evaluation index of the cost.

E. REPRESENTATION OF ACTIONS INVOLVING STATE TRANSITION OF ENVIRONMENT

The feasible actions for the robot vary after the execution of the action involving the state transition of the environment, such as the change in the object placement. Therefore, the actions involving the state transition of the environment are represented by directed edges, which connect the action possibility graphs before and after the state transition.

First, the possible state transitions of the environment are understood based on the properties of the actions associated with objects. Then, the posterior action possibility graphs after the recognized possible state transition $G^a_{1,1\sim N_1^O}$ are constructed. N_1^O is the number of possible environmental states in a single transition from the initial environment state O_0. Finally, the action graph is constructed by connecting the prior possibility graph G_0^a and each posterior possibility graph $G^a_{1,1\sim N_1^O}$ with edges representing the actions that cause the state transition. These procedures represent not only the *movement* action but also actions involving the state transition of the environment, which are associated with the objects in the environment based on the subsystem configuration and prior knowledge, as a graph in the robot configuration space.

The expression method of actions involving the state transition is described below in detail.

1) UNDERSTANDING POSSIBLE ENVIRONMENTAL STATE

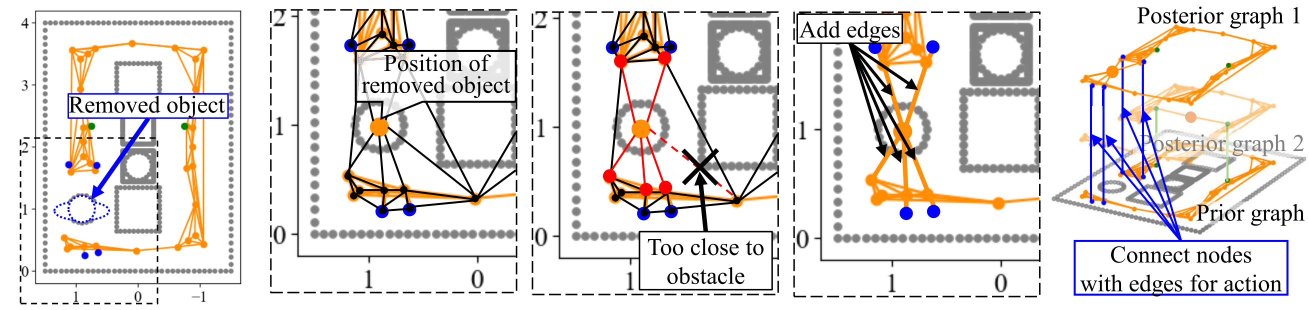
All possible environment states $O_{1,1\sim N^{O_1}}$ in a single transition from the initial environment state O_0 are captured. Specifically, the possible states $O_{1,1\sim N^{O_1}}$ are captured as combinations of the influenced object o_e^a and effect e^a based on the actions a associated with the objects and the action property given as prior knowledge, as shown in Table 2.

2) CONSTRUCTION OF ACTION POSSIBILITY GRAPH AFTER ACTIONS INVOLVING STATE TRANSITION OF ENVIRONMENT

The posterior action possibility graphs $G^a_{1,1\sim N^{O_1}}$ are constructed for each recognized possible state transition of the environment, which is the combination of the influenced object o_e^a and effect e^a. It is assumed that the action effect e^a based on prior knowledge has no uncertainty in the graph construction.

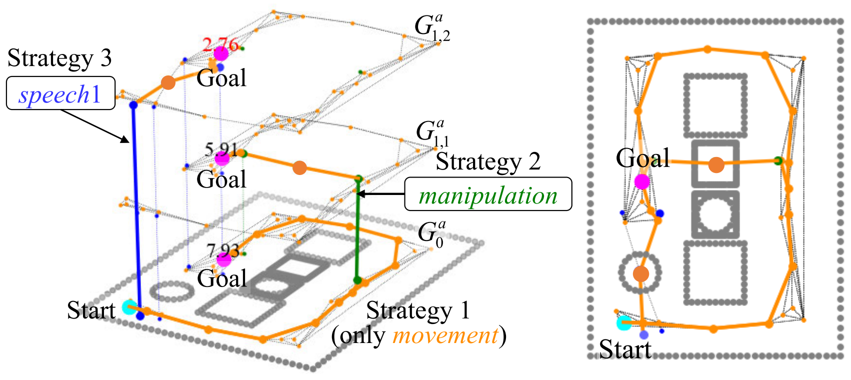
In general, the posterior action possibility graph $G^a_{1,n}$ is constructed by adding and deleting relevant nodes and edges from the prior action possibility graph G_0^a based on the initial environment state O_0, the influenced object o_e^a, and the effect e^a. We describe a simpler approach to construct a posterior action possibility graph $G^a_{1,n}$ for *removed* as an example of action effects e^a. *remove* means that the influenced object o_e^a is removed, and the object's position is traversable. Therefore, to demonstrate that the position of the influenced object o_e^a has become traversable, the posterior action possibility graph $G^a_{1,n}$ is constructed by adding the node and edges to the prior action possibility graph G_0^a.

First, a new node is added to the position of the influenced object o_e^a in the prior action possibility graph G_0^a. In addition, to identify the nodes reachable to the new node for the robot, we applied Delaunay triangulation to the nodes in the prior action possibility graph G_0^a and the new node, as shown in Fig. 5b. The candidates of the reachable nodes are the nodes connected to the position of the influenced object o_e^a in the result of Delaunay triangulation. Then, the reachable nodes are selected from the candidates by checking the reachability, whether the robot can reach from the candidate's position to the new node's position considering the surrounding obstacles considering the robot's footprint, as shown in Fig. 5c. Finally, the posterior action possibility graph $G^a_{1,n}$ is obtained by connecting the new nodes to the selected reachable nodes with new edges, as shown in Fig. 5d. The action label of the added edges l^a is set to *movement*. In addition, the cost of the added edges c^a is calculated based on the length of the edges and the evaluation index of the cost.



(a) Prior action possibility graph **(b)** Delaunay triangulation **(c)** Selecting reachable nodes **(d)** Obtained posterior action possibility graph **(e)** Action representation

FIGURE 5. Construction of action possibility graph after state transition of environment and Representation of actions involving state transition.



(a) Perspective view **(b)** Top view

FIGURE 6. Task planning based on action graph.

The posterior action possibility graphs $G^a_{1,1\sim N^{o_1}}$ are constructed by applying the above procedures for each combination of the influenced object o^a_e and effect e^a.

3) CONNECTING ACTION POSSIBILITY GRAPH BEFORE AND AFTER THE STATE TRANSITION

The action graph G^a is constructed by connecting the prior action possibility graph G^a_0 and the posterior action possibility graphs $G^a_{1,1\sim N^{o_1}}$ with the edges. Specifically, the nodes related to the action causing the state transition of the environment in the prior and posterior action possibility graphs are connected to each other by directed edges, as shown in Fig. 5e. These edges represent the actions involving the state transitions of the environment and are set with the attributes and costs of the actions. Also, the costs are calculated based on the action content and evaluation index.

IV. ACTION-BASED SPATIO-TEMPRAL ROBOT NAVIGATION

A. OVERVIEW

Action-based spatio-tempral robot navigation (ASTRON) is a task planning method in real space for the navigation tasks. The navigation tasks are instructed with a pair of positions corresponding to the start and the goal in a two-dimensional absolute coordinate system.

In ASTRON, the action sequence that effectively utilizes the robot's functions is derived by interpreting the task within the scope of the action possibility based on the action graph. The flowchart of ASTRON is as shown in Fig. 1. The system operates through the process of environmental recognition, task planning, and action execution. In this study, we focus on the task planning, which consists of problem representation, setting, and planning, and the proposed components corresponds to the problem representation and setting. First, for the representation focusing on the action possibility of the robot, the action graph is automatically constructed by the procedure described in section III. Then, the action sequence to reach the goal from the start is derived by applying Dijkstra's method to the constructed action graph. The obtained action sequence contains the necessary information to execute in real space because it consists of the positions (nodes) and the contents (edges) of the feasible actions based on the action graph.

The details of the task planning based on the action graph are described below.

B. APPLYING ACTION GRAPH TO TASK PLANNING FOR ROBOT NAVIGATION

1) SETTING

It is necessary to determine the start and goal nodes corresponding to the instructed start and goal positions to apply

TABLE 3. Verification contents.

Exp. No.	Simulation or Real	Verification items: Subsystem-level affordance	Geometric reasoning	Multi-layer	Feasibility	Effective use of functions	Scenario No.	Env. No.	Robot No.
1	Simulation	○	○				1.1	1	3
							1.2		4
2	Simulation			○			\	1	4
3	Real				○	○	3.1	2	1
							3.2		2
							3.3		3
							3.3	3	4

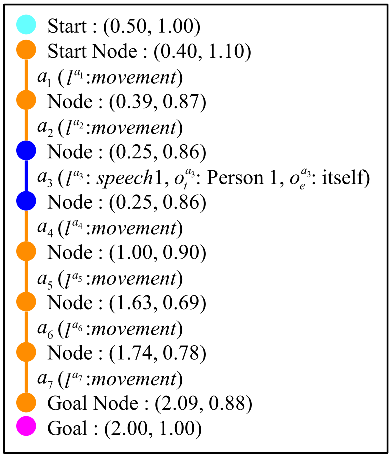


FIGURE 7. Example of action sequence (Strategy 3).

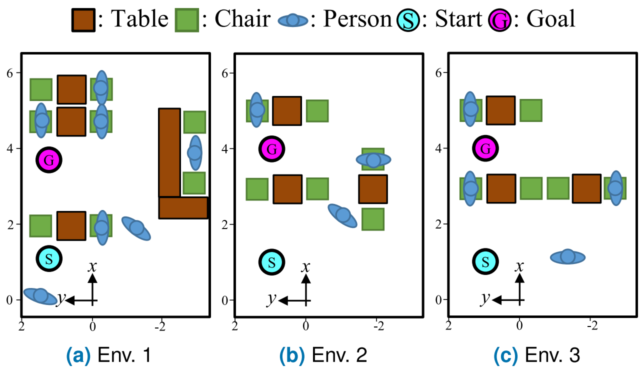


FIGURE 8. Experimental environments.

the action graph to task planning. Because the action graph consists of prior and posterior action possibility graphs to the state transition of the environment caused by the action, there are multiple nodes in the action graph that indicate the same position. However, the strategy, or the necessary action, for the robot to the position is different in the nodes. Therefore, the start node is set to the node nearest to the initial robot's position in the prior action possibility graph G_0^a, in the task planning based on the action graph. In addition, the goal nodes are set to the nodes nearest to the goal position in the prior and posterior action possibility graphs G_1^a.

For example, in the environment as shown in section III, the goal nodes are set to three nodes, as shown in Fig. 6: The first one is in the prior action possibility graph G_0^a, which corresponds to the strategy to reach the goal with only the *movement* action. The second one is in the first posterior action possibility graph $G_{1,1}^a$, which corresponds to the strategy of interfering with the chair by the *movement* action and the *manipulation* action. The third is in the second posterior action possibility graph $G_{1,2}^a$, which corresponds to the strategy of talking to a person with the *movement* action and the *speech* action.

2) PLANNING

Then, the optimal action sequence A to reach the goal from the start is derived by applying an optimization algorithm to the action graph. In this study, Dijkstra's method is adopted as an example of fundamental algorithms for finding the shortest paths between nodes in a graph, and other methods can also be applied. First, Dijkstra's method is applied to the pairs of start and goal nodes for every strategy that the robot can choose. Then, the action sequences with the least cost of each strategy are obtained as shown in Fig. 6. Finally, the action sequence with the least cost among the obtained action sequences is selected as the optimal one A. The cost function of the actions can be designed based on the movement distance, required time, energy consumption, etc. In this study, the cost is calculated based on the movement distance as an example.

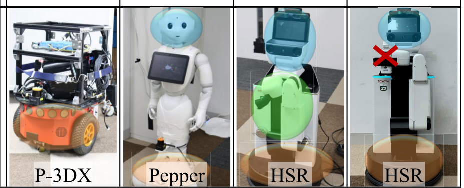
The action sequence A derived in the example problem shown in Fig. 6 is shown in Fig. 7.

V. EXPERIMENT

A. OVERVIEW

The verification contents are presented in Table 3. In Exp. 1, we conducted an ablation study with simulations to verify the understanding of the robot's action possibility considering both subsystem-level affordance and geometric reasoning. In Exp. 2, we verified that an action graph with multi-layered action possibility graphs can represent a wide variety of actions and can capture the available strategies through an ablation study with simulations. In Exp. 3, we conducted actual machine experiments on several robots with different subsystem configurations to verify that the proposed method

TABLE 4. Robot list.

Robot No.		1	2	3	4
Robot					
action	*movement*	○	○	○	○
	*speech*1	×	○	○	○
	*speech*2	×	×	×	○
	manipulation	×	×	○	×

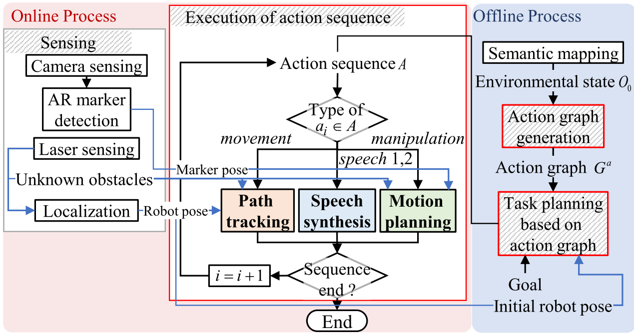


FIGURE 9. System configuration of ASTRON.

could plan feasible action sequences that could effectively use the functions of the robot.

B. EXPERIMENTAL SETTING

1) ENVIRONMENT

The simulations and the actual machine experiments were conducted in environments simulating a coffee shop, as shown in Fig. 8. We assumed that the robot and the staff cooperate with each another in the shop, and the standing people represent the shop's staff who cooperate with the robot.

2) ROBOT

These experiments were conducted on four different subsystem configurations with three different robots, as shown in Table 4. Robot 1 is a mobile robot called P-3DX [23], with a mobile base as its available subsystem. Robot 2 is a mobile communication robot called Pepper [24], with a mobile base with built-in and a speaker as its available subsystems. Robot 3 is a mobile manipulator called HSR [25], which has a mobile base, a speaker, and a manipulator. Robot 4 is also the HSR, which grasps an object using the manipulator and can perform the *speech*2 action instead of the *manipulation* action. As for the sensors, all robots are equipped with laser range finders, and a camera is available for Robot 3.

3) SYSTEM CONFIGURATION

The system configuration is a design example of a robot navigation system based on ASTRON as shown in Fig. 9. In addition, the specific values of the parameters are shown

TABLE 5. System parameters.

Process	Parameter		Value
Action graph generation	Affordance range of *speech*2	$r^{speech2}$	$0.5 \sim 2.0$ m
	Appropriate distance for *speech*1, 2	d^s	0.5 m
	Appropriate distance for *manipulation*	d^m	1.0 m
Task planning based on action graph	Evaluation index of cost		Distance
	c^a of *movement*		Edge length
	c^a of *speech*1, 2		0.3 m
	c^a of *manipulation*		0.3 m

in Table 5. The robots recognized objects in the environment based on the action graph and the action sequence planned based on the graph, and the obstacles using the mounted laser range finder. It was assumed that the environmental state did not change significantly during the short period of the navigation task.

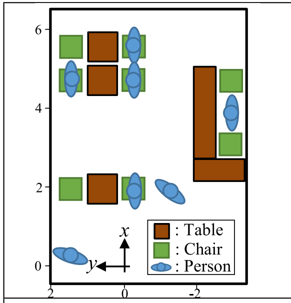
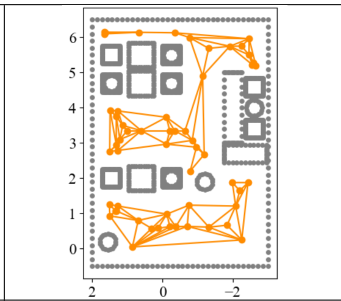
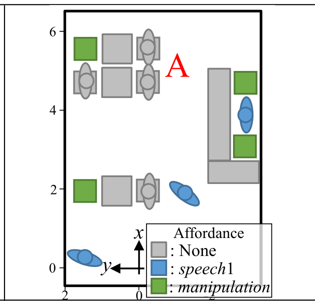
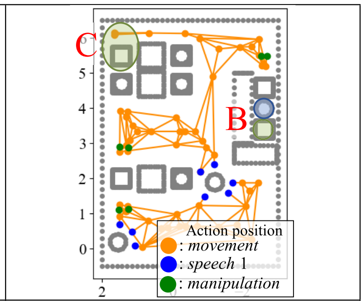
In the offline process, the system first obtains the environmental state O_0 based on a pre-constructed semantic map. The class, position, and size of the objects in the environment are obtained by using an environmental sensing system, which estimates by applying the object instance segmentation algorithm [26] to images from 12 RGB-D cameras [27]. Then, the action graph G^a is constructed based on the obtained environmental state O_0, the prior knowledge shown in Tables 1 and 2, and the parameters shown in Table 5. The environmental sensing system was used for more effective verification rather than for the construction of the action graph. Finally, the action sequence A is obtained through task planning based on the action graph. The evaluation index of action cost c^a is the distance travelled by the robot. Therefore, the cost is set considering the necessary distance to execute the action, as shown in Table 5, and as follows

$$c^a = \begin{Bmatrix} \|\boldsymbol{p}^{a+1} - \boldsymbol{p}^a\| & (a = movement) \\ 0.3 & (a = speech1, 2) \\ 0.3 & (a = manipulation) \end{Bmatrix}, \tag{4}$$

where $\boldsymbol{p}^a$ and $\boldsymbol{p}^{a+1}$ are the starting position and the ending position of the *movement* action, individually. Specifically, the *movement* cost is the edge length, and the cost of the *speech*1, 2 actions and the *manipulation* action are set considering that these actions involve only a slight movement

In the online process, the robot performs actions on the obtained action sequences. Each action a_i is processed depending on the action label l^{a_i} and is executed; here, i is the action index of the action sequences. The robot's velocity in the *movement* action was calculated based on the fuzzy potential method, which enabled the integrated consideration of unknown obstacles observed using the laser range sensor and the position of the next node [28]. In the *manipulation*

TABLE 6. Result of Scenario 1.1 in Exp. 1.

<table>
<tr>
<td>
</td>
<td></td>
<td>
</td>
<td>
</td>
</tr>
<tr><td>Environment</td><td>Com. 1.1: Geometric reasoning</td><td>Com. 1.2: Action association</td><td>Pro: Action possibility graph G_0^a</td></tr>
<tr><td>Geometric reasoning</td><td>○</td><td></td><td>○</td></tr>
<tr><td>Subsystem-level affordance</td><td></td><td>○</td><td>○</td></tr>
</table>

action, the robot planned and followed the motion to move towards the table based on the camera sensing of the AR marker installed on the chair.

C. EXPERIMENT 1: VERIFICATION OF ACTION POSSIBILITY GRAPH

1) EXPERIMENTAL CONDITIONS

In Exp. 1, we conduct verification under two scenarios in which the environment is Env. 1 contains various object placements, and the robots are Robot 3 and 4 which have different subsystem configurations, as shown in Table 3, Exp. 1.

2) RESULT

The result is shown in Tables 6 and 7. The figure of Com. 1.1 in Tables 6 and 7 shows the action possibility graph for movement G^m obtained by applying the geometric reasoning described in section III-B. The orange nodes and edges are related to the *movement* action.

The figures of Com. 1.2 in Tables 6 and 7 show the result of the action association with each object based on the environmental state O_0 according to the subsystem-level affordance. In Scenario 1.1, the *speech*1 action or the *manipulation* action are associated with objects because the assumed robot is Robot 3. Meanwhile, the *speech*1, 2 actions are associated with the objects in Scenario 1.2, where the target robot is Robot 4. In particular, the area marked A in the figure shows the features of the action associations based on the positional relationships between the objects. Specifically, the actions are not associated with people sitting on the chairs. In addition, when the *speech*2 action is feasible, the *speech*2 action is associated with pairs of a person and a chair within a certain distance, but not to the chairs outside the range. Thus, we confirmed that the actions could be associated with objects according to the robot's subsystem configuration and the arrangement of the object set for task planning in real space.

The figures of Pro. in Tables 6 and 7 show the construction result of the action possibility graph G^a considering both the geometric reasoning and the action association described above. Additionally, the blue and green nodes indicate the execution positions of the *speech*1 and *manipulation* actions, respectively, in Scenario 1.1. In Scenario 1.2, the blue nodes indicate the execution positions of the *speech*1, 2 actions. In particular, the areas marked B and C in the figure show the features of the action possibility understanding based on both geometric reasoning and action association. In the areas marked B, the objects are associated with actions, but the nodes for action positions are not set because it was found based on the geometric reasoning described in section III-D1 that the robot could not reach there. Additionally, there are no nodes around the objects associated with the actions because the geometric reasoning described in section III-D2 revealed that the robot could not maintain an appropriate distance from the object for the action. Thus, we confirmed that the execution position of feasible actions in real space could be realized by constructing an action possibility graph that considers both geometric reasoning and action association.

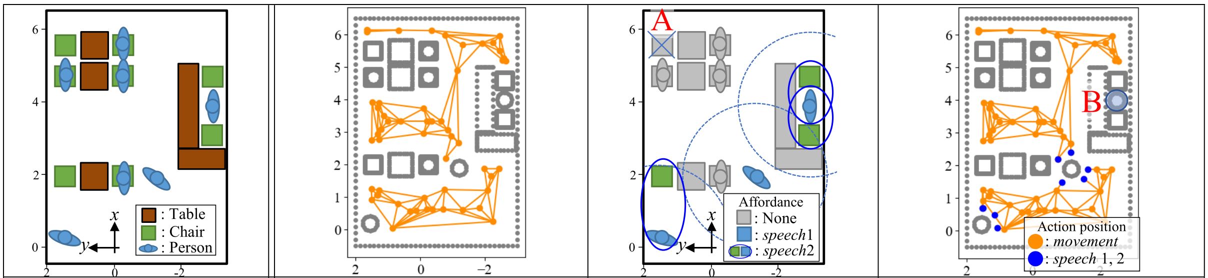
D. EXPERIMENT 2: VERIFICATION OF ACTION GRAPH STRUCTURE

1) EXPERIMENTAL CONDITIONS

In this experiment, the assumed environment was Env. 1, and the assumed robot was Robot 4, as with Scenario 1.2 in Exp. 1. The action possibility graph for movement G^m and the result of the action association were the same as those of Com. 1.1 and Com. 1.2 in Table 7.

Action graphs were constructed based on the two comparisons and the proposed graph representation methods. Then, the action sequences based on each action graph were compared. The conditions of the graph representation methods are shown in Table 8. The comparison method 2.1 represents the action possibility in the current environment as well as the methods to understand the objects to be removed in real space [8], [12]. In this method, the movable object positions were traversable, and the cost of removing the object was added when the robot passes through the position. The comparison

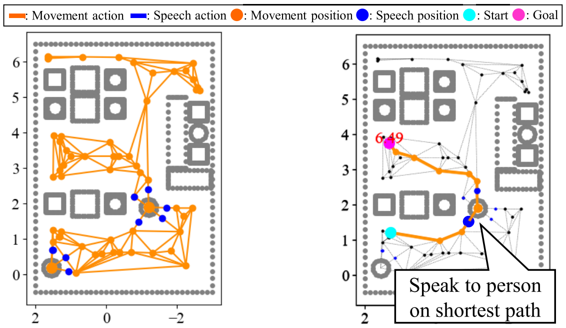
TABLE 7. Result of Scenario 1.2 in Exp. 1.



Environment	Com. 1.1: Geometric reasoning	Com. 1.2: Action association	Pro: Action possibility graph G_0^a
Geometric reasoning	○		○
Subsystem-level affordance		○	○

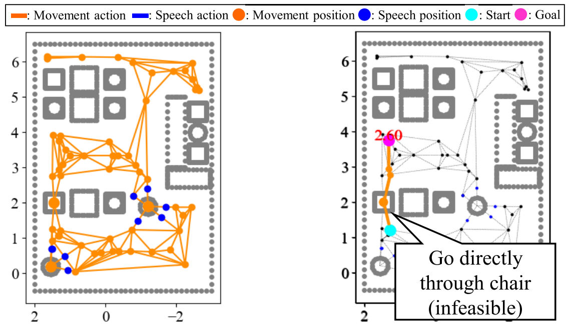
TABLE 8. Condition of graph representation methods in Exp. 2.

Method / Contents	Com. 2.1	Com. 2.2	Pro. Action graph
prior action possibility G_0^a	○		○
posterior action possibility G_1^a		○	○



(a) Graph (b) Result of planning

FIGURE 10. Result of Com.2.1 in Exp. 2.



(a) Graph (b) Result of planning

FIGURE 11. Result of Com.2.2 in Exp. 2.

method 2.2 represents the change of the action possibility caused by actions, that is, the posterior action possibility. The positions that the robot can make traversable by its actions were set as passable, and the cost of the actions was added to the *movement* cost of the edges connected to nodes for actions other than the *movement* action.

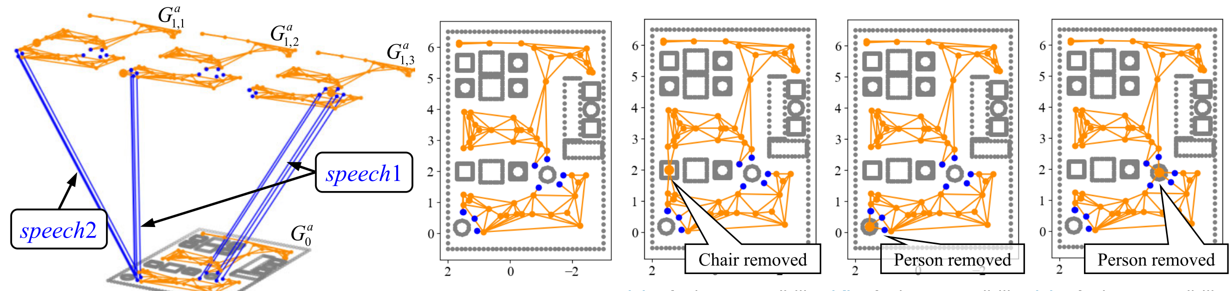
2) RESULT

The results are shown in Fig. 10-13. Fig. 10a shows the obtained graph based on the method Com. 2.1, which represents the action possibility for the current state of the environment, that is, the prior action possibility. The action sequence obtained by applying Dijkstra's method to the obtained graph is shown in Fig. 10b. The plan enables the robot to ask the person blocking its way to let the robot through and move toward the goal.

The graph shown in Fig. 11a is obtained based on the method Com. 2.2, which represents the action possibility for the environment state after the transition, that is, the posterior action possibility. In addition, the obtained action sequence based on the graph is shown in Fig. 11b. The obtained sequence implies that the robot passes through the position of the chair associated with the *speech*2 and approaches the goal. However, because the graph does not specify what action causes the state transition of the environment, the obtained sequence does not include asking the person to remove the chair, and thus is not feasible for the robot.

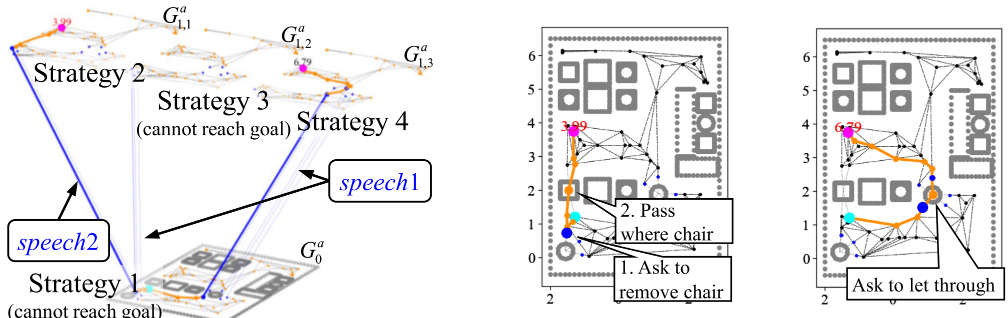
The obtained action graph based on the proposed method that represents the prior and posterior action possibility graphs to the state transition of the environment is shown in Fig. 12a. The planning result based on the obtained action graph is shown in Fig. 13. Four selectable strategies are obtained because there are four possible environments, including no state transitions. No action sequence is obtained for strategy 1 in which the robot makes no state transition, and strategy 3, in which the robot asks the person at (0.2, 1.5) to let through, as shown in Fig. 12d. The obtained sequence for strategy 2 makes the robot to first approach the person at (0.2, 1.5), perform the *speech*2 action in order to ask him to remove the chair at (2.0, 1.4), and then pass through the chair space. The obtained sequence for strategy 4 is the same as that based on Com. 2.1.

The costs of action sequences obtained based on each method are summarized in Table 9. The action sequence with the least cost is that obtained based on Com 2.2, but



(a) Action graph (b) Action possibility graph G^a_0 (c) Action possibility graph $G^a_{1,1}$ (d) Action possibility graph $G^a_{1,2}$ (e) Action possibility graph $G^a_{1,3}$

FIGURE 12. Action graph constructed based on proposed method in Exp. 2.

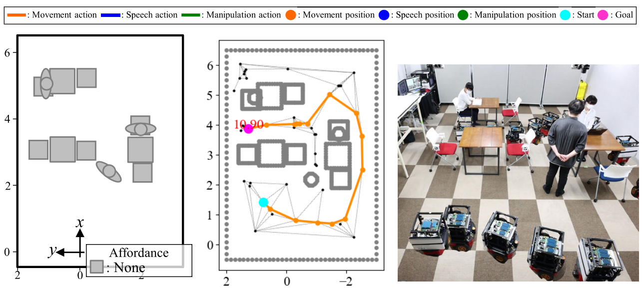


(a) Overview of planning result (b) Plan of strategy 2 (c) Plan of strategy 4

FIGURE 13. Planning result based on action graph based on proposed method in Exp. 2.

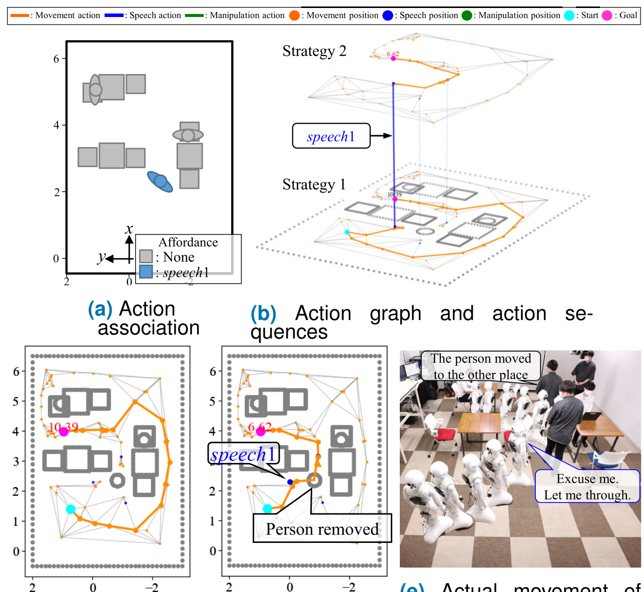
TABLE 9. Cost of planned action sequence in Exp. 2.

Com 2.1	Com 2.2	Proposed			
		Strat. 1	Strat. 2	Strat. 3	Strat. 4
6.79	2.60 (infeasible)	N/A	3.89	N/A	6.79



(a) Action association (b) Action graph and action sequence (c) Actual movement

FIGURE 14. Result of Scenario 3.1 in Exp. 3.



(a) Action association (b) Action graph and action sequences (c) Strategy 1 (d) Strategy 2 (e) Actual movement of strategy 2

FIGURE 15. Result of Scenario 3.2 in Exp. 3.

it is infeasible. Based on the proposed method, the action sequence for strategy 2 has the least cost in the executable action sequences. This solution is an action sequence that includes the *speech*2 action that is difficult to represent in Com 2.1.

Thus, we confirmed that for more diverse actions, it is necessary for action graphs to clearly describe the action possibilities and the changes in them caused by the action, and the multi-layered action possibility graphs can accurately represent the available strategies.

E. EXPERIMENT 3: ACTUAL MACHINE EXPERIMENT

1) EXPERIMENTAL CONDITIONS

In this experiment, we conducted experiments in which the actual robot works following the planned action sequence from the start to the goal using the system Fig. 9. The conditions of each scenario are shown in Table 3, Exp. 3. In Scenario 3.1-3.3, Env. 3 and three types of robots, Robot 1-3 with different subsystem configura-

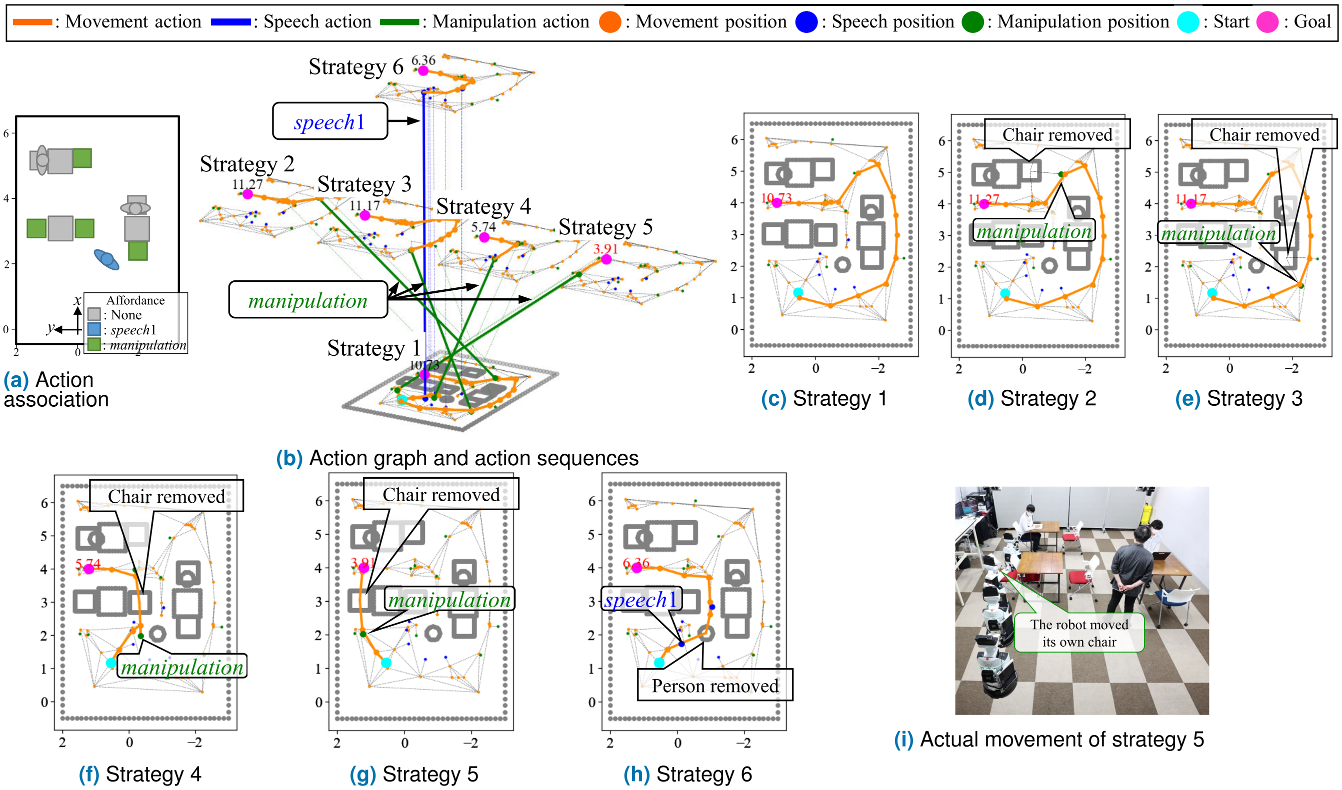


FIGURE 16. Result of Scenario 3.3 in Exp. 3.

tions were targeted. Also, Robot 4 worked in Env. 4 for Scenario 3.4.

2) RESULTS

The results of Scenario 3.1-3.3 are shown in Figs. 14-16. The results of the action association are shown in Figs. 14a, 15a and 16a. It was confirmed that different actions were associated with objects depending on the subsystem configuration of each robot, even in the same environment. The action graph and the planning result for each scenario are shown in Figs. 14b, 15b and 16b. The action sequences for the selectable strategies are shown in Figs. 15c, 15d and 16c to 16h.

In Scenario 3.1, the planned action sequence was to approach the goal by avoiding all the objects as obstacles because the mobile robot has only the mobile base, as shown in Fig. 14a and 14b. The strategy corresponded to strategy 1 in Scenario 3.1 and 3.2. On the other hand, because the target robot also had a speaker in Scenario 3.2, the selectable strategy included not only the strategy to avoid all the objects but also to ask a person to let through, as shown in Figs. 15a-15d. Furthermore, in Scenario 3.3, where the robot had a speaker and a manipulator, an action graph in which the *speech*1 and the *manipulation* actions are associated with objects was constructed, as shown in Figs. 15a-15d. The selectable strategies were six strategies consisting of a strategy to avoid all the objects, to ask a person, and to remove each of the four movable chairs. The costs of the action sequences for the strategies derived in Scenario 3.1-3.3 are summarized in Table 10.

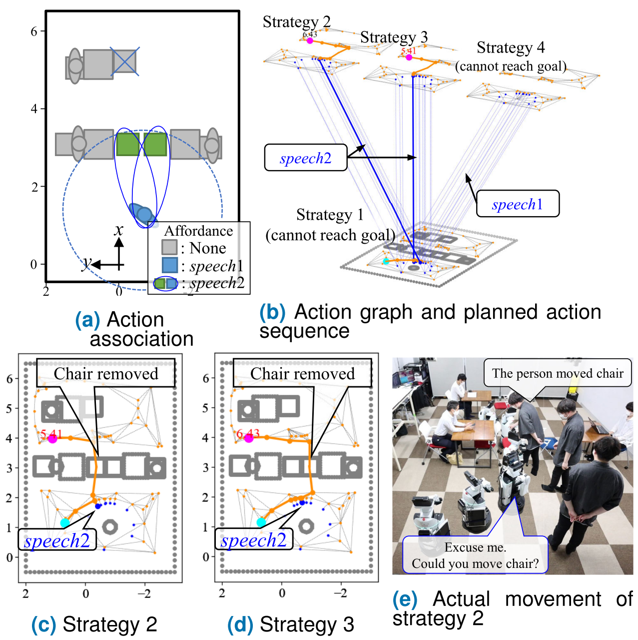
TABLE 10. Cost of the action sequences for each strategies in Scenario 3.1 to 3.3 of Exp. 3.

: Movement only : With speech : With manipulation

Scenario No.	Strat. 1	Strat. 2	Strat. 3	Strat. 4	Strat. 5	Strat. 6
3.1	10.90	\				
3.2	10.39	6.62	\			
3.3	10.73	11.27	11.17	5.74	3.91	6.36

Note here that the environmental states are slightly different in each scenario because of the observation noise of the environmental sensing system. Figs. 14c, 15e and 16i show the actual robot's performance in real space following the least cost action sequence among the possible choices for each scenario. In Scenario 3.1, the robot reached the goal without contact with any objects. In addition, the robot went to speak to the person blocking the way, asked him to give way, and then reached the goal in Scenario 3.2. Further, in Scenario 3.3, the robot approached the chair, removed the chair by itself with the manipulator, and reached the destination. The above results show that the proposed method can obtain a feasible action sequence that can effectively utilize the robot's subsystems among the available strategies, depending on its subsystem configuration.

The results of Scene 3.4 are shown in Fig. 17. The *speech*1, 2 actions were associated based on the positional relation between a person and a chair in the action association as shown in Fig. 17a. The derived strategies included the strategy to ask a person with the *speech*1, 2 actions as shown



(a) Action association (b) Action graph and planned action sequence (c) Strategy 2 (d) Strategy 3 (e) Actual movement of strategy 2

FIGURE 17. Result of Scenario 3.4 in Exp. 3.

TABLE 11. Cost of planned action sequences in Scenario 3.4 of Exp. 3.

Strat. 1	Strat. 2	Strat. 3	Strat. 4
N/A	5.41	6.43	N/A

in Fig. 17b-17d. The costs of the action sequences for each strategy are summarized in Table 11. No action sequence was obtained for strategy 1 to avoid all objects and strategy 4 to ask a person to let through because the robot could not reach the goal. The result of the action sequence for strategy 2, for which the cost was least, is shown in Fig. 17e. Following the action sequence, the robot actively approached the person, requested him to remove the chair, and then passed through it to reach the destination. The results of this scenario show that the planned sequence is feasible in real space, as it can represent actions where the target object o_t^a and the influenced object o_e^a do not match.

In addition, the position to speak to a person was determined to make the distance traveled the shortest among where the robot could speak to a person by considering the movement after speaking, as shown in Fig. 17c. This is another feature of task planning in real space where object placement can be considered.

VI. CONCLUSION

The planner to exploit the functions that each robot has to the fullest is necessary to achieve the task provided by a user. The objective of this study was to enable the robot to derive a feasible action sequence for effective use of the functions by interpreting the instruction within the action possibility of the robot in real space and the change in it.

In this study, an action graph was first proposed as a novel environmental representation to facilitate the understanding of the robot's action possibility in real space. The action graph represents the robot's action possibility and the change in it according to both the subsystem-level affordance and the geometric reasoning related to the robot and the object set in the environment, where the edge and the nodes represent the action executions their specific positions, respectively. Then, an action-based spatio-temporal robot navigation (ASTRON) was proposed, which is a task planning method in real space that focuses on robot navigation. ASTRON enabled the robot to obtain the action sequence to reach a goal with more effective use of its functions by understanding the selectable strategies, that is, the action sequences for each possible state transition of the environment based on the action graph.

In the experiments, we first confirmed that the action possibility graph constructed according to both the subsystem-level affordance and the geometric reasoning facilitates the understanding of the action possibility. In addition, it was confirmed that the action graph consisting of multiple action possibility graphs facilitates the representation of various actions and the understanding of the selectable strategies in the environment. In the actual machine experiments with multiple robots with different subsystem configurations, we verified that the proposed method enables the robots to obtain a feasible action sequence to utilize their subsystems.

In this study, it was assumed that people in the environment were cooperative with the robots. The proposed method also assumed that the environmental state would not change dynamically in the short task period. We plan to extend the proposed method for handling unexpected situations out of these assumptions in future research. One concrete approach is the online update of the action graph and the action sequence considering the uncertainty based on the partial observation using the sensors on the robot. Then, it is necessary to analyze the extended method statistically, such as the success rate, to evaluate the handling of the unexpected situations.

REFERENCES

[1] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," *Artif. Intell.*, vol. 2, no. 3, pp. 189–208, 1971.
[2] M. Fox and D. Long, "PDDL2.1: An extension to PDDL for expressing temporal planning domains," *J. Artif. Intell. Res.*, vol. 20, pp. 61–124, Dec. 2003.
[3] S. Edelkamp and J. Hoffmann, "PDDL 2.2: The language for the classical part of the 4th international planning competition," Albert Ludwigs-Universitat Freiburg, Institut für Informatik, Tech. Rep. 195, 2004.
[4] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5079–5085.
[5] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robot. Auto. Syst.*, vol. 66, pp. 86–103, Apr. 2015.
[6] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4921–4933, Dec. 2020.
[7] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.
[8] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 639–646.
[9] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "An incremental constraint-based framework for task and motion planning," *Int. J. Robot. Res.*, vol. 37, no. 10, pp. 1134–1151, Sep. 2018, doi: 10.1177/0278364918761570.

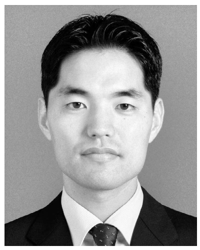
[10] N. T. Dantam, S. Chaudhuri, and L. E. Kavraki, ''The task-motion kit: An open source, general-purpose task and motion-planning framework,'' *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, pp. 61–70, Sep. 2018.
[11] H.-N. Wu, M. Levihn, and M. Stilman, ''Navigation among movable obstacles in unknown environments,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1433–1438.
[12] M. Wang, R. Luo, A. O. Onol, and T. Padir, ''Affordance-based mobile robot navigation among movable obstacles,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2734–2740.
[13] H. Moravec and A. Elfes, ''High resolution maps from wide angle sonar,'' in *Proc. IEEE Int. Conf. Robot. Automat.*, Mar. 1985, pp. 116–121.
[14] N. Rhinehart and K. M. Kitani, ''Learning action maps of large environments via first-person vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 580–588.
[15] R. Hachiuma, Y. Ozasa, and A. Yorozu, ''Socially adaptive manner for motion planning from human server in cafe,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Workshop (IROSW)*, Oct. 2018, pp. 1–2.
[16] F.-J. Chu, R. Xu, L. Seguin, and P. A. Vela, ''Toward affordance detection and ranking on novel objects for real-world robotic manipulation,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4070–4077, Oct. 2019.
[17] T. Lüddecke, T. Kulvicius, and F. Wörgötter, ''Context-based affordance segmentation from 2D images for robot actions,'' *Robot. Auto. Syst.*, vol. 119, pp. 92–107, Sep. 2019.
[18] S. Bazeille and D. Filliat, ''Incremental topo-metric SLAM using vision and robot odometry,'' in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 4067–4073.
[19] J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez, ''A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM,'' in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Apr. 2007, pp. 2061–2067.
[20] P.-T. Wu, C.-A. Yu, S.-H. Chan, M.-L. Chiang, and L.-C. Fu, ''Multi-layer environmental affordance map for robust indoor localization, event detection and social friendly navigation,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2945–2950.
[21] C. R. G. Dreher, M. Wachter, and T. Asfour, ''Learning object-action relations from bimanual human demonstration using graph networks,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 187–194, Jan. 2020.
[22] M. J. Aein, E. E. Aksoy, and F. Wörgötter, ''Library of actions: Implementing a generic robot execution framework by using manipulation action semantics,'' *Int. J. Robot. Res.*, vol. 38, no. 8, pp. 910–934, Jul. 2019, doi: 10.1177/0278364919850295.
[23] MobileRobots. (Aug. 2021). *P-3DX Product Page*. [Online]. Available: https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf
[24] Softbank. (Aug. 2021). *Pepper Product Page*. [Online]. Available: https://www.softbank.jp/robot/pepper/
[25] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, ''Development of human support robot as the research platform of a domestic mobile manipulator,'' *ROBOMECH J.*, vol. 6, no. 1, pp. 1–15, Dec. 2019.
[26] K. He, G. Gkioxari, P. Dollar, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[27] Microsoft. (Aug. 2021). *Azure Kinect DK Product Page*. [Online]. Available: https://azure.microsoft.com/ja-jp/services/kinect-dk/
[28] T. Suzuki and M. Takahashi, ''Obstacle avoidance for autonomous mobile robots based on position prediction,'' in *Proc. IEEE Int. Conf. Control, Automat. Robot.*, Jul. 2009, pp. 299–304.

YOSUKE KAWASAKI (Member, IEEE) received the B.S. degree from the Department of System Design Engineering, Keio University, Yokohama, Japan, in 2018, and the M.S. degree in engineering from the School of Science for Open and Environmental Systems, Keio University, in 2020.

His research interests include learning from demonstration, service robot, and human–robot interaction.

SHUNSUKE MOCHIZUKI received the B.S. degree from the Department of System Design Engineering, Keio University, Yokohama, Japan, in 2020. He is currently a Graduate Student.

His primary research interests include intelligent robotics, path planning, and environment recognition.



MASAKI TAKAHASHI (Member, IEEE) received the B.S. degree from the Department of System Design Engineering, Keio University, Yokohama, Japan, in 2000, and the M.S. and Ph.D. degrees in engineering from the School of Science for Open and Environmental Systems, Keio University, in 2002 and 2004, respectively.

From 2005 to 2008, he worked as a Research Assistant at the Department of System Design Engineering, Keio University, where he became an Associate Professor, in 2009, and a Professor, in 2019. His primary research interests include human–robot interaction, motion and vibration control, and sensor fusion. He is a member of the American Institute of Aeronautics and Astronautics, the Japan Society of Mechanical Engineers, and the Robotics Society of Japan.