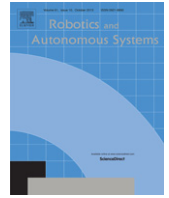




Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Service robot system with an informationally structured environment



Yoonseok Pyo^{a,*}, Kouhei Nakashima^a, Shunya Kuwahata^a, Ryo Kurazume^b,
Tokuo Tsuji^b, Ken'ichi Morooka^b, Tsutomu Hasegawa^c

^a Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motoooka, Nishi-ku, Fukuoka-shi, Fukuoka 819-0395, Japan

^b Faculty of Information Science and Electrical Engineering, Kyushu University, 744 Motoooka, Nishi-ku, Fukuoka-shi, Fukuoka 819-0395, Japan

^c Kumamoto National College of Technology, Koshi-shi, Kumamoto 861-1102, Japan

HIGHLIGHTS

- Introduction of architecture and components of the ROS-TMS.
- Integration of various data from distributed sensors for service robot system.
- Object detection system (ODS) using RGB-D camera.
- Motion planning for a fetch-and-give task using a wagon and a humanoid robot.
- Handing over an object to a human using manipulability of both a robot and a human.

ARTICLE INFO

Article history:

Received 6 October 2014

Received in revised form

27 April 2015

Accepted 10 July 2015

Available online 21 July 2015

Keywords:

Informationally structured environment

Intelligent space

Distributed sensor

Service robot

Robot operating system

Motion planning

ABSTRACT

Daily life assistance is one of the most important applications for service robots. For comfortable assistance, service robots must recognize the surrounding conditions correctly, including human motion, the position of objects, and obstacles. However, since the everyday environment is complex and unpredictable, it is almost impossible to sense all of the necessary information using only a robot and sensors attached to it. In order to realize a service robot for daily life assistance, we have been developing an informationally structured environment using distributed sensors embedded in the environment. The present paper introduces a service robot system with an informationally structured environment referred to the ROS-TMS. This system enables the integration of various data from distributed sensors, as well as storage of these data in an on-line database and the planning of the service motion of a robot using real-time information about the surroundings. In addition, we discuss experiments such as detection and fetch-and-give tasks using the developed real environment and robot.

© 2015 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Aging of the population is a common problem in modern societies, and rapidly aging populations and declining birth rates have become more serious in recent years. For instance, the manpower shortage in hospitals and elderly care facilities has led to the deterioration of quality of life for elderly individuals. Robot technology is expected to play an important role in the development of a healthy and sustainable society. In particular, daily life assistance for elderly individuals in hospitals and care facilities is one of the most urgent and promising applications for service robots.

For a service robot, information about its surrounding, such as the positions of objects, furniture, humans, and other robots is indispensable for safely performing proper service tasks. For example, in order to deliver an object requested by a user, a robot must know the location of the target object and which trajectory should be followed to reach the object. However, current sensing technology, especially for cases of robots equipped with external sensors, is not good enough to complete these tasks satisfactorily. For example, a vision system is susceptible to changes in lighting conditions and the appearances of objects. Moreover, the field of vision is rather narrow. Although occlusions can be partly solved by sensors on a mobile robot, background changes and unfavorable vibrations of a robot body make processes more difficult. In addition, the payload of a robot is not so high and computer resources are also limited.

* Corresponding author. Tel.: +81 92 802 3605.

E-mail address: pyo@irvs.ait.kyushu-u.ac.jp (Y. Pyo).

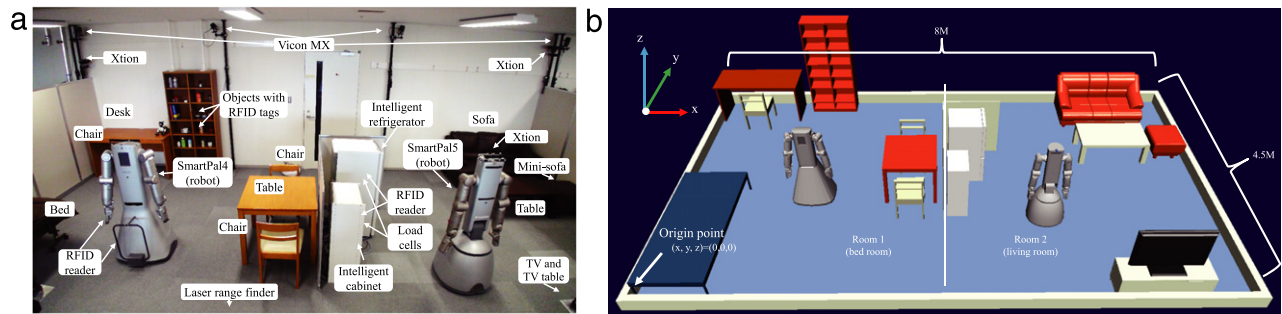


Fig. 1. An informationally structured room: (a) arrangement of furniture, robots, and sensors, (b) coordinates system and room dimensions.

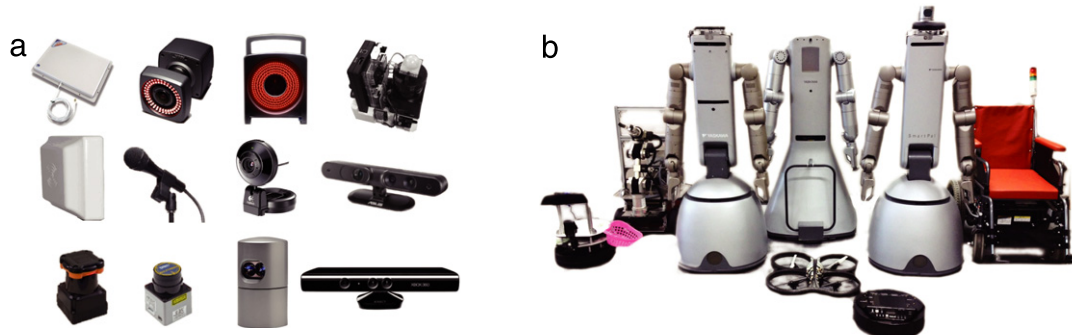


Fig. 2. Distributed sensors and service robots: (a) distributed sensors, (b) service robots.

On the other hand, fixed sensors in an environment are more stable and can more easily gather information about the environment. If a sufficient number of sensors can be embedded in the environment in advance, occlusion is no longer a crucial problem. Information required to perform tasks is acquired by distributed sensors and transmitted to a robot on demand. The concept of making an environment smarter rather than the robot is referred to as an informationally structured environment [1–6].

An informationally structured environment is a feasible solution for introducing service robots into our daily lives using current technology, and several systems that observe human behavior using distributed sensor systems and provide proper service tasks according to requests from human or emergency detection, which is triggered automatically, have been proposed [1,7–12]. Several service robots that act as companions to elderly people or as assistants to humans who require special care have been developed [13–18].

We also have been developing an informationally structured environment for assisting in the daily life of elderly people in our research project, i.e., the Robot Town Project [2,19]. The goal of this project is to develop a distributed sensor network system covering a town-size environment consisting of several houses, buildings, and roads, and to manage robot services appropriately by monitoring events that occur in the environment. Events sensed by an embedded sensor system are recorded in the Town Management System (TMS), and appropriate information about the surroundings and instructions for proper services are provided to each robot [2].

We also have been developing an informationally structured platform (Fig. 1) in which distributed sensors (Fig. 2a) and actuators are installed to support an indoor service robot (Fig. 2b). For example, in this platform, objects such as books, pens, pet bottles, chairs, and desks are detected by embedded sensors and RFID tags, and all of the data are stored in the TMS database. A service robot performs various service tasks according to the environmental data stored in the TMS database in collaboration with distributed sensors and actuators, for example, installed in a refrigerator to open a door.

We herein introduce a new Town Management System called the ROS–TMS. In this system, the Robot Operating System (ROS [20]) is adopted as a communication framework between various modules, including distributed sensors, actuators, robots, and databases. Thanks to the ROS, we were able to develop a highly flexible and scalable system. Adding or removing modules such as sensors, actuators, and robots, to or from the system is simple and straightforward. Parallelization is also easily achievable.

We herein report the followings:

- Introduction of architecture and components of the ROS–TMS
- Object detection using a sensing system of the ROS–TMS
- Fetch-and-give task using the motion planning system of the ROS–TMS.

The detection and fetch-and-give tasks are among the most frequent tasks for elderly individuals in daily life and are a typical application task for the proposed system.

The remainder of the present paper is organized as follows. After presenting related research in Section 2, we introduce the architecture and components of the ROS–TMS in Section 3. In Section 4, we describe the sensing system of the ROS–TMS for processing the data acquired from various sensors. Section 5 describes the robot motion planning system of the ROS–TMS used to design the trajectories for moving, gasping, giving, and avoiding obstacles using the information on the environment acquired by the sensing system. We present the experimental results for service tasks performed by a humanoid robot and the ROS–TMS in Section 6. Section 7 concludes the paper.

2. Related research

A considerable number of studies have been performed in the area of informationally structured environments/spaces to provide human-centric intelligent services. Informationally structured environments are referred to variously as home automation systems, smart homes, ubiquitous robotics, kukanchi,

and intelligent spaces, depending on the field of research and the professional experience of the researcher.

Home automation systems or smart homes are popular systems that centralize the control of lighting, heating, air conditioning, appliances, and doors, for example, to provide convenience, comfort, and energy savings [21–23]. The informationally structured environment can be categorized in this system, but the system is designed to support not only human life but also robot activity for service tasks.

Hashimoto and Lee proposed an intelligent space in 1996 [24]. Intelligent spaces (iSpace) are rooms or areas that are equipped with intelligent devices, which enable spaces to perceive and understand what is occurring within them [24–27]. These intelligent devices have sensing, processing, and networking functions and are referred to as distributed intelligent networked devices (DINDs). One DIND consists of a CCD camera to acquire spatial information and a processing computer, which performs data processing and network interfacing. These devices observe the position and behavior of both human beings and robots coexisting in the iSpace.

Moreover, the concept of a physically embedded intelligent system (PEIS) has been introduced in 2005 [28]. A PEIS involves the intersection and integration of three research areas: artificial intelligence, robotics, and ubiquitous computing. Anything that consists of software components with a physical embodiment and interacts with the environment through sensors or actuators/robots is considered to be a PEIS, and a set of interconnected physically embedded intelligent systems is defined as a PEIS ecology. Inside a PEIS ecology, the software components should interact, communicate, and achieve goals together. Tasks can be achieved using either centralized or distributed approaches using the PEIS ecology [29,30].

Ubiquitous robotics [31] involves the design and deployment of robots in smart network environments in which everything is interconnected. The authors define three types of Ubibots: software robots (Sobots), embedded robots (Embots), and mobile robots (Mobots), which can provide services using various devices through any network, at any place and at any time in a ubiquitous space (u-space). Embots can evaluate the current state of the environment using sensors, and convey that information to users. Mobots are designed to provide services and explicitly have the ability to manipulate u-space using robotic arms. A Sobot is a virtual robot that has the ability to move to any location through a network and to communicate with humans. The present authors have previously demonstrated the concept of a PIES using Ubibots in a simulated environment and u-space [32,33].

In Europe, RoboEarth [34–36] is essentially a World Wide Web for robots, namely, a giant network and database repository in which robots can share information and learn from each other about their behavior and their environment. The goal of RoboEarth is to allow robotic systems to benefit from the experience of other robots, paving the way for rapid advances in machine cognition and behavior, and ultimately, for more subtle and sophisticated human–machine interactions.

The informationally structured environment/space (also referred to as Kukanchi, a Japanese word meaning interactive human–space design and intelligence [37,38]) has received a great deal of attention in robotics research as an alternative approach to the realization of a system of intelligent robots operating in our daily environment. Human-centered systems require, in particular, sophisticated physical and information services, which are based on sensor networks, ubiquitous computing, and intelligent artifacts. Information resources and accessibility within an environment are essential for people and robots. The environment surrounding people and robots should have a structured platform for gathering, storing, transforming, and providing information. Such

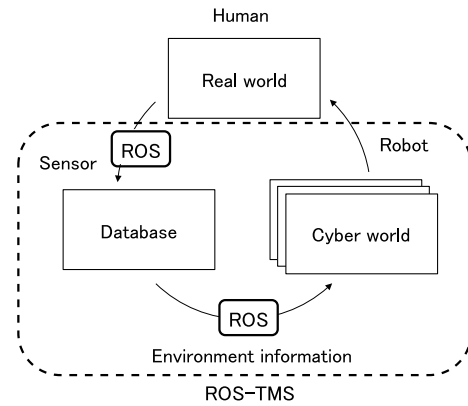


Fig. 3. Conceptual diagram of the ROS-TMS.

an environment is referred to as an informationally structured space by the IEEE Symposium on Robotic Intelligence in Informationally Structured Space (RiSS [39]).

In Section 5, we present a coordinate motion planning technique for a fetch-and-give including handing over an object to a person. The problem of handing over an object between a human and a robot has been studied in Human–Robot Interaction (HRI) [40–43]. In particular, the work that is closest to ours is the one by Dehais et al. [42]. In their study, physiological and subjective evaluation for a handing over task was presented. The performance of hand-over tasks were evaluated according to three criteria: legibility, safety and physical comfort. These criteria are represented as fields of cost functions mapped around the human to generate ergonomic hand-over motions. Although their approach is similar to our approach, we consider the additional criteria, that is, the manipulability of both a robot and a human for a comfortable and safety fetch-and-give task.

The problem of pushing carts using robots has been reported in many studies so far [44–50]. The earlier studies in pushing a cart were reported using a single manipulator mounted on a mobile base [44,45]. In these systems, a single manipulator held a cart at a single point, and the planning of effector force to produce desired trajectories was discussed. The problem of towing a trailer has also been discussed as an application of a mobile manipulator and a cart [46]. This work is close to the approach in this paper, however, a pivot point using a cart is placed in front of the robot in our technique.

The work that is closest to ours is the one by Scholz et al. They provided a solution for real time navigation in a cluttered indoor environment using 3D sensing [49]. Though the first attempt with the cart by Tan et al. [44,45] was limited to simple paths using an open-loop controller, Scholz et al. proposed the solution to execute smooth and arbitrary trajectories in a closed loop controller with PR2. Many previous works focus on the navigation and control problems for movable objects. On the other hand, we consider the problem including handing over an object to a human using a wagon, and propose a total motion planning technique for a fetch-and-give task with a wagon.

3. Overview of the ROS-TMS

We have been developing an informationally structured architecture referred to as the Town Management System (TMS) for assisting in the daily life of elderly people [2]. In the present paper, we extend the TMS and develop a new Town Management System called the ROS-TMS.

A conceptual diagram of the ROS-TMS is shown in Fig. 3. This system has three primary components, i.e., real-world, database, and cyber-world components. Events occurring in the real world,

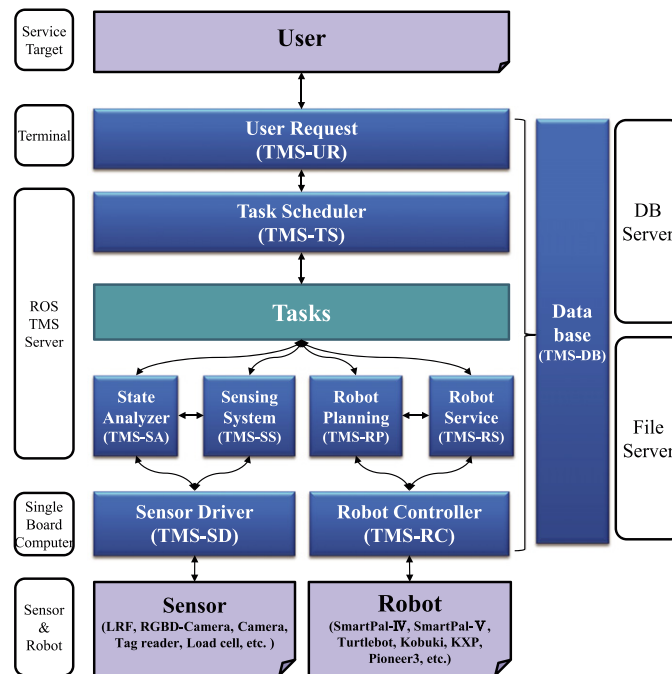


Fig. 4. Architecture of the ROS-TMS.

such as user behavior or user requests, and the current situation of the real world, such as the positions of objects, humans, and robots are sensed by a distributed sensing system. The gathered information is then stored in the database. Appropriate service commands are planned using the environmental information in the database and are simulated carefully in the cyber world using simulators, such as choreonoid [51]. Finally, service tasks are assigned to service robots in the real world.

In order to realize the above-described concept, the following functions are implemented in the ROS-TMS.

1. Communication with sensors, robots, and databases.
2. Storage, revision, backup, and retrieval of real-time information in an environment.
3. Maintenance and providing information according to individual IDs assigned to each object and robot.
4. Notification of the occurrence of particular predefined events, such as accidents.
5. Task schedule function for multiple robots and sensors.
6. Human–system interaction for user requests.
7. Real-time task planning for service robots.

The ROS-TMS has unique features such as high scalability and flexibility, as described below.

- **Modularity:** The ROS-TMS consists of 73 packages categorized into 11 groups and 151 processing nodes. Re-configuration of structures, for instance adding or removing modules such as sensors, actuators, and robots, is simple and straightforward owing to the high flexibility of the ROS architecture.
- **Scalability:** The ROS-TMS is designed to have high scalability so that it can handle not only a single room but also a building and a town.
- **Diversity:** The ROS-TMS supports a variety of sensors and robots. For instance, Vicon MX (Vicon Motion Systems Ltd.), TopUrg (Hokuyo Automatic), Velodyne 32e (Velodyne Lidar), and Oculus Rift (Oculus VR) are installed in the developed informationally structured platform (Fig. 1) for positioning and human–system interaction. SmartPal-IV and V (Yaskawa Electric Corp.), AR Drone (Parrot), TurtleBot2 (Yujin Robot Co., Ltd.), a wheelchair robot, and a mobile robot with a 5-DOF

manipulator are also provided in order to accomplish a variety of service tasks.

- **Safety:** Data gathered from the real world is used to perform simulations in the cyber world in order to evaluate the safety and efficiency of designed tasks. According to the simulation results, proper service plans are designed and provided by service robots in the real world.
- **Privacy protection:** One important restriction in our intelligent environment is to install a small number of sensors to avoid interfering with the daily activity of people and to reduce the invasion of their privacy as far as possible. For this reason, we do not install conventional cameras in the environment. Although the robots are equipped with several cameras, these cameras are only used in certain restricted scenarios.
- **Economy:** Sensors installed in an environment can be shared with robots and tasks, and thus we do not need to equip individual robots with numerous sensors. In addition, most sensors are processed by low-cost single-board computers in the proposed system. This concept has an advantage especially for the system consisting of multiple robots since robots can share the resources in the environment. Currently, the cost of a distributed sensor network may be higher than the cost of a single robot. However, we believe that distributed sensor network (or Internet of Things, IoT) will become a standard facility in our society in the near future, and the cost of a sensor network will be greatly reduced since all the devices will be designed to be connected each other.

Some features mentioned above such as modularity, scalability, and diversity owe much to ROS's outstanding features. On the other hand, economical or processing efficiency strongly depends on the unique features of ROS-TMS, since various information gathered by distributed sensor networks is structured and stored to the database and repeatedly utilized for planning various service tasks by robots or other systems.

Fig. 4 shows the overall architecture of ROS-TMS. This system is composed of five components: user, sensor, robot, task, and data. These components are also composed of sub-modules, such as the User Request sub-module for the user component, the Sensor Driver sub-module, the Sensing System and State Analyzer

Table 1
Components of the ROS-TMS.

Component	Node name	Description
Data	Database (TMS_DB)	TMS_DB is composed of the database, DB_Writer, which writes the information, DB_Reader, which performs information readout, and DB_manager, which manages the stored data. Implementation of the database itself is performed using MySQL, and, currently, information is stored as follows: <ul style="list-style-type: none"> • Registered information about the robots, such as their shape and appearance • Information related to the sensors embedded in the environment • Static environment information, such as maps • Dynamic information, such as the calculated position of moving objects • Task information classified by robot services
User	User Request (TMS_UR)	TMS_UR can receive service requests from users using remote controllers, smartphones, and tablets, as well as other portable devices. These requests are delivered to TMS_TS. Fig. 5 shows some examples of TMS_UR including a smart phone, a tablet, and wearable glasses.
Task	Task Scheduler (TMS_TS)	Execution management is performed for robot services delivered via TMS_UR. The task Scheduler also has features for robot service execution, suspension, resuming, and rescheduling after errors.
Sensor	Sensor Driver (TMS_SD)	TMS_SD is a package for obtaining raw data from the sensor. We use various sensors, such as an LRF, an RGB-D camera, a USB camera, a Velodyne LiDAR sensor, and a Vicon motion capture.
	Sensing System (TMS_SS)	By interpreting the data acquired from various environment sensors, more precise information concerning the environment can be obtained. TMS_SS consists of a floor sensing system (FSS) for detecting objects on the floor using a laser range finder, an intelligent cabinet system (ICS) to detect the type and position of objects through RFID tag readers and load cells on the cabinet, motion tracking using Xtion and Kinect, location tracking of moving objects using a Vicon camera, and object detection using RGB-D cameras mounted on robots (ODS).
	State Analyzer (TMS_SA)	Based on information acquired from TMS_DB, there are nodes that perform task command generation, estimation of the current state, and detection. The results of these estimations are then transmitted to TMS_TS as execution requests for robot services. We have implemented an object tracking system that maintains a record of the objects that have been moved/placed by people. TMS_SA detects when a person removes objects according to the intelligent cabinet and estimates which objects have a probability to be left unmoved based on the walking trajectories of people.
Robot	Robot Motion Planning (TMS_RP)	Robots can perform operation planning in order to successfully avoid obstacles based on information acquired from TMS_DB. Furthermore, the output is converted into primitive behaviors for the robots, such as moving, grasping, and giving. Robot services can be then provided according to the service behavior descriptions based on the combination of the above-mentioned features.
	Robot Service (TMS_RS)	A node group that performs services that do not require motion programming, such as tuning on a TV, opening a door, and capturing an image.
	Robot Controller (TMS_RC)	Nodes to control the robots using a planned path. We use robots such as SmartPal-IV and V (Yaskawa Electric), Turtlebot2, Kobuki, a wheelchair robot, and custom robots.

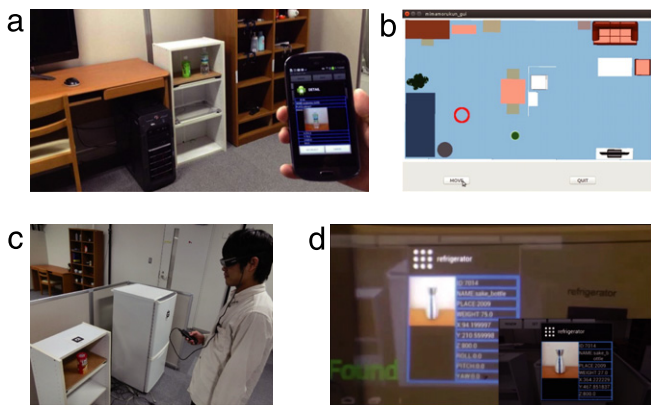


Fig. 5. Examples of TMS_UR: (a) object search system using a smartphone, (b) navigation system for a wheelchair robot using a tablet, (c and d) a smart glasses (Moverio, Epson) to display object information using AR technology.

sub-modules for the sensor component, the Robot Controller, the Robot Motion Planning, and the Robot Service sub-modules for the robot component, the Task Scheduler sub-module for the task component, and the Database sub-module for the data component. Brief explanations of each node group are presented in Table 1. The details of these components will be described later herein.

4. Sensing system

The sensing system (TMS_SS) is a component of the ROS-TMS that processes the data acquired from various environment sensors. In the current platform (Fig. 1), TMS_SS is composed of three sub-packages as described below and is described in Table 1.

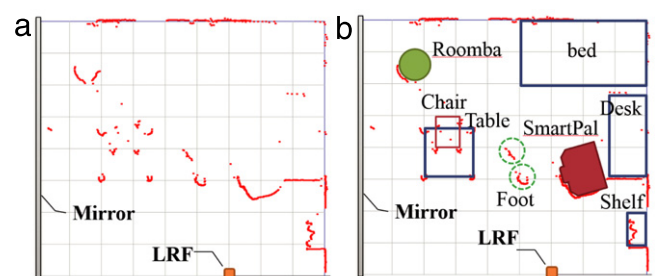


Fig. 6. Floor sensing system used to identify objects: (a) original data points, (b) identification of objects.

1. Floor sensing system (FSS)
2. Intelligent cabinet system (ICS)
3. Object detection system (ODS)

4.1. Floor sensing system (FSS)

The current platform (Fig. 1) is equipped with a floor sensing system to detect objects on the floor and people walking around. This sensing system is composed of a laser range finder located on one side of the room and a mirror installed along another side of the room. This configuration allows a reduction of dead angles of the LRF and is more robust against occlusions [52]. An example of object detection using this system is shown in Fig. 6. People tracking is performed by first applying static background subtraction and then extracting clusters in the remainder of the measurements. Clusters are later tracked by matching profiles of cluster corresponding to legs and extending the motion using the



Fig. 7. Information about objects provided by the intelligent cabinet. Squares on the screen indicate the position of the different objects together with their description.



Fig. 8. SmartPal-V is equipped with two RGB-D cameras.

accelerations of the legs. Moreover, this system can measure the poses of the robot and movable furniture such as a wagon using tags, which have encoded reflection patterns optically identified by the LRF [53].

4.2. Intelligent cabinet system (ICS)

The cabinets installed in the room are equipped with RFID readers and load cells to detect the types and positions of the objects in the cabinet. Every object in the environment has an RFID tag containing a unique ID that identifies it. This ID is used to retrieve the attributes of the object, such as its name and location in the database. Using the RFID readers, we can detect the presence of a new object inside the cabinet. In addition, the load cell information allows us to determine its exact position inside the cabinet. An example of object detection in the intelligent cabinet is shown in Fig. 7. Additional details about the intelligent cabinets used herein can be found in [54].

4.3. Object detection system (ODS)

If neither the FSS nor the ICS is available for detecting objects such as those placed on a desk, the object detection system using a RGB-D camera on a robot is provided in this platform as shown in Fig. 8. In this system, a newly appeared object or movement of an object is detected as a change in the environment. The steps of the change detection process are as follows.

1. Identification of furniture
2. Alignment of the furniture model
3. Object extraction by furniture removal
4. Segmentation of objects
5. Comparison with the stored information

4.3.1. Identification of furniture

It is possible to identify furniture based on the position and posture of robots and furniture in the database. Using this information, robot cameras determine the range of the surrounding environment that is actually being measured. Afterwards, the system superimposes these results and the position information for furniture to create an updated furniture location model.

The point cloud (Fig. 9a) acquired from the robot is superimposed with the furniture's point cloud model (Fig. 9b). The results are shown in Fig. 9c. After merging the point cloud data (Fig. 9a) and the point cloud model (Fig. 9b), as shown in Fig. 9c, the system deletes all other points except for the point cloud model for the furniture and limits the processing range from the upcoming steps.

4.3.2. Alignment of the furniture model

We scan twice for gathering point cloud datasets of previous and current scene. In order to detect the change in the newly acquired information and stored information, it is necessary to align two point cloud datasets obtained at different times because these data are measured from different camera viewpoints. In this method, we do not try to directly align the point cloud data, but rather to align the data using the point cloud model for the furniture. The reason for this is that we could not determine a sufficient number of common areas by simply combining the camera viewpoints from the two point cloud datasets and can also reduce the amount of information that must be stored in memory. Using the aligned point cloud model, it is possible to use the point cloud data for objects located on the furniture, without having to use the point cloud data for furniture from the stored data. The alignment of the furniture model is performed using the ICP algorithm.

4.3.3. Object extraction by furniture removal

After alignment, all points corresponding to furniture are removed to extract an object. When removing the furniture according to the method described in a previous study using voxelized shape and color histograms [55], it is particularly difficult to cleanly remove items of furniture such as beds, blankets, and pillows, because they are easily deformable by hand or other objects. As such, the system removes furniture according to segmentation using color information and three-dimensional positions. More precisely, the point cloud is converted to a RGB color space and then segmented using a region-growing method. Each of the resulting segments is segmented based on the XYZ space. Using this method, each of these last segments becomes a continuous area of the same color. The system then selects only those segments that overlap with the model and then removes these segments. Fig. 10 shows the results of bed removal after blankets have been changed.

4.3.4. Segmentation of objects

After performing the above-mentioned processing, only the points associated with objects placed on furniture remain. These points are further segmented based on XYZ space. The resulting segments are stored in the database.

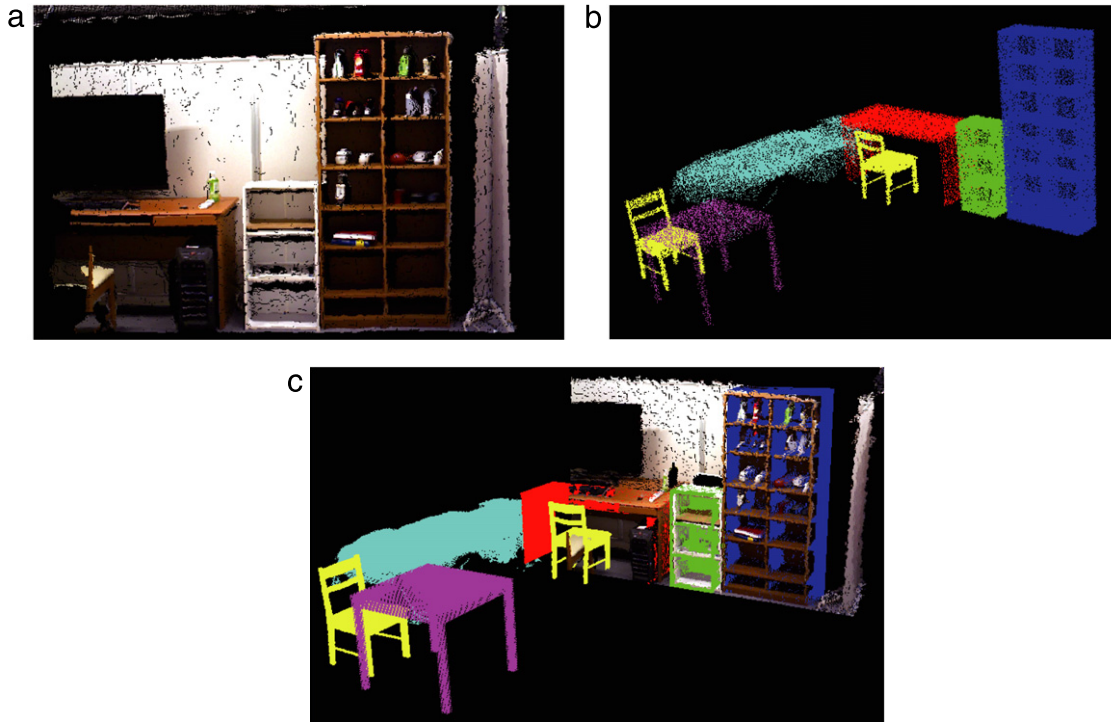


Fig. 9. Superposition of the acquired point cloud data and the point cloud model for the furniture: (a) acquired point cloud data, (b) point cloud model for the furniture, (c) overlay of acquired point cloud data and point cloud model for the furniture.

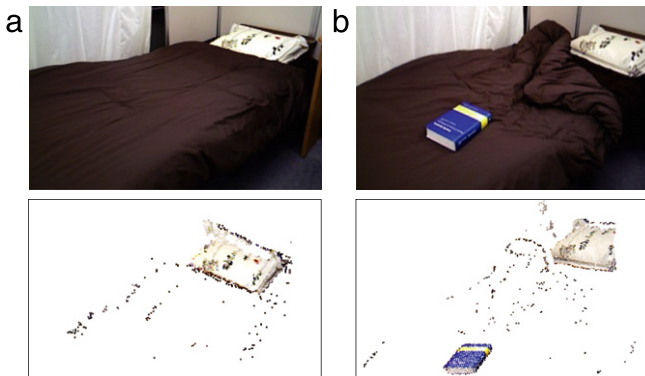


Fig. 10. Bed removal after changing the blanket: (a) memory information, (b) newly acquired information.

4.3.5. Comparison with the stored information

Finally, the system associates each segment from the previously stored information with the newly acquired information. As a result, the system finds the unmatched segments and captures the movement of objects that has occurred since the latest data acquisition. In other words, the segments that did not match between the previous dataset and the newly acquired dataset, reflect objects that were moved, assuming that the objects were included in the previously stored dataset. On the other hand, the segments that appear in the most recent dataset, but not in the previously stored dataset, reflect objects that were recently placed on the furniture. The set of segments that are included in the association process are determined according to the center position of segments. For the segments sets from the previous dataset and the newly acquired dataset, the association is performed based on a threshold distance between their center positions, considering the shape and color of the segments as the arguments for the association. We use an elevation map (Fig. 11b) that describes the height of furniture above the reference surface level to represent the shape of the object.

The reference surface level of furniture is, more concretely, the top surface of a table or shelf, the seat of a chair, or the sleeping surface of a bed. The elevation map is a grid version of the reference surface level and is a representation of the vertical height of each point with respect to the reference surface level on each grid.

When various vertical height measurements occur within a single grid, the chosen measurement value corresponds to the highest point from the reference surface. Then, comparison is performed on the elevation map for each segment, taking into consideration the variations in size, the different values obtained from each grid, and the average value for the entire map. The color information used to analyze the correlation between segments is the hue (H) and saturation (S). A two-dimensional histogram for H and S is created for each segment, as shown in Fig. 11c. The number of bins for H and S is 32. Using these H–S histograms, the previous data and the newly acquired data are compared, allowing the system to determine whether it is dealing with the same objects. The Bhattacharyya distance $BC(p, q)$ within H–S histograms (p, q) is used for determining the similarity between histograms and is calculated according to Eq. (1).

$$BC(p, q) = \sum_{h=0}^{31} \sum_{s=0}^{31} \sqrt{p(h, s)q(h, s)} \quad (1)$$

Once distance values are calculated, the object can be assumed to be the same as for the case in which the degree of similarity is equal to or greater than the threshold value. According to the scenarios for the two tables shown in Fig. 12a and b, the objects placed on top of the tables becomes segmented, and the correlation between these segments is calculated using only H–S histograms. Fig. 12c shows the Bhattacharyya distance of the H–S histograms according to the overall combination of the previously stored dataset and the newly acquired dataset. If the distance is the highest for both sectors and is equal to or greater than the threshold, the system determines it to be the same object. The distances shown in bold represent the values for which it is regarded as the same object.

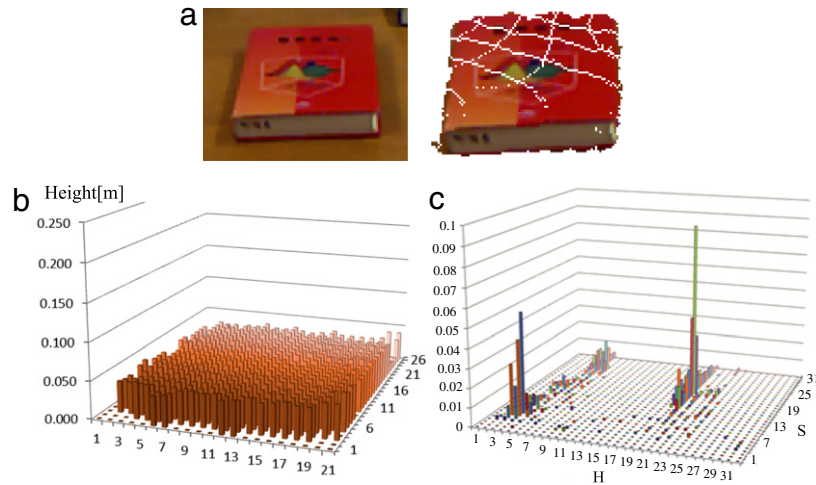
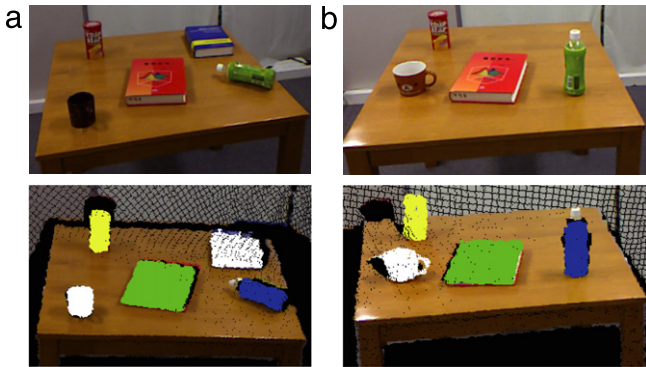


Fig. 11. Elevation-map and H-S histogram: (a) target object and obtained point cloud data, (b) elevation-map, (c) H-S histogram.



		New data			
		Object1	Object2	Object3	Object4
Previous data					
Object1		0.918	0.078	0.394	0.619
Object5		0.164	0.269	0.195	0.263
Object2		0.069	0.837	0.137	0.161
Object6		0.541	0.075	0.544	0.591
Object4		0.580	0.150	0.561	0.850

Fig. 12. Comparison of H-S histograms: (a) memory information, (b) newly acquired information, (c) Bhattacharyya distance.

5. Robot motion planning

Robot motion planning (TMS_RP) is the component of the ROS-TMS that calculates the movement path of the robot and the trajectories of the robot arm for moving, grasping, giving, and avoiding obstacles based on information acquired from TMS_SS and is described in Table 1.

We consider the necessary planning to implement services such as fetch-and-give tasks because such tasks are among the

most frequent tasks required by elderly individuals in daily life. Moreover, robot motion planning includes wagons for services that can carry and deliver a large amount of objects, for example, at tea time or handing out towels to residents in elderly care facilities as shown in Fig. 14a.

Robot motion planning consists of sub-planning, integration, and evaluation of the planning described below to implement the fetch-and-give task.

1. Grasp planning to grip a wagon
2. Position planning for goods delivery
3. Movement path planning
4. Path planning for wagons
5. Integration of planning
6. Evaluation of efficiency and safety

Each planning, integration, and evaluation process uses environment data obtained from TMS_DB and TMS_SS, so that no further actions are required in order to acquire data. Moreover, actions that integrate overall planning are obtained from a combination of various individual planning threads. The integration method is considered to be not only efficient but also safe in places such as hospitals and elderly care houses. The output consists of a series of actions that can be executed efficiently and safely.

5.1. Grasp planning to grip a wagon

In order for a robot to push a wagon, the robot needs to grasp the wagon at first. There is an infinite number of base positions that the robot can have relative to a wagon that the robot must grip. However, a robot can push a wagon in a stable manner if the robot grasps the wagon from two poles positioned on its sides. Thus, the number of base position options for the robot with respect to the wagon is reduced to four (i) as shown in Fig. 14. The position and orientation of the wagon, as well as its size, is managed using the ROS-TMS database. Using this information, it is possible to determine the correct relative position. This provides the distance, i.e., the control distance (CD), between the robot and the wagon when the robot is actually grasping the wagon. Based on the wagon direction when the robot is grasping its long side, valid candidate points can be determined using Eqs. Eq. (2) through (4) below ($i = 0, 1, 2, 3$). Here, R represents the robot, and W represents the wagon. Subscripts x , y , and θ represent the corresponding x -coordinate, y -coordinate, and posture (rotation of the z -axis). Fig. 13 shows the positional relationship between the robot and the

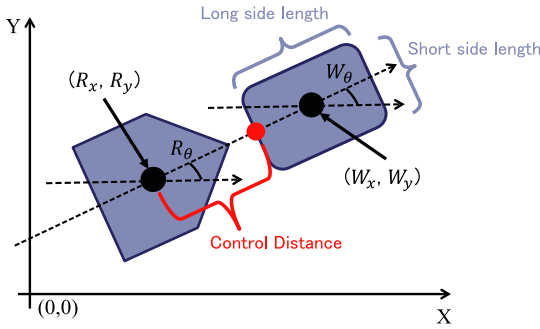


Fig. 13. Base position of the robot at the time of gripping the wagon ($i = 2$).

wagon, given $i = 2$. Moreover, Fig. 14 shows the wagon gripping position as a 3D model, given $W_\theta = 0$.

$$R_{x_i} = W_x + \left(\frac{W_{\text{size}_i}}{2} + CD \right) \cos \left(W_\theta + \frac{i}{2}\pi \right) \quad (2)$$

$$R_{y_i} = W_y + \left(\frac{W_{\text{size}_i}}{2} + CD \right) \sin \left(W_\theta + \frac{i}{2}\pi \right) \quad (3)$$

$$R_{\theta_i} = W_\theta + \pi + \frac{i}{2}\pi \quad (4)$$

$$W_{\text{size}_i} = \begin{cases} \text{length of the wagon's long side} & (i = 0, 2) \\ \text{length of the wagon's short side} & (i = 1, 3) \end{cases}$$

5.2. Position planning for goods delivery

In order to hand over goods to a person, it is necessary to plan both the position of the goods to be delivered and the base position of the robot according to the person's position. Using manipulability as an indicator for this planning, the system plans the position of the goods relative to the base position. Manipulability is represented by the degree to which hands/fingers can move when each joint angle is changed. When trying to deliver goods in postures with high manipulability, it is easier to modify the motion, even when small gaps exist between the robot and the person. Since it is difficult to know the exact position of the person, or to operate the robot without errors, a method that deals with these gaps is indispensable. The velocity vector \mathbf{v} corresponds to the position of hands, and \mathbf{q} is the joint angle vector. In addition, we assume the high manipulability of the arm of the person makes him more comfortable for grasping goods. Their relation is represented in Eqs. (5) and (6).

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (5)$$

$$\omega = \sqrt{\det \mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q})} \quad (6)$$

If the arm has a redundant degree of freedom, an infinite number of joint angle vectors corresponds to just one hand position. Therefore, when solving the inverse kinematics of this issue, we calculate the posture that represents the highest manipulability within the range of possible joint angle movements.

The planning procedure for the position of goods and the position of robots using manipulability is as follows:

1. The system maps the manipulability that corresponds to the robots and each person on the local coordinate system.
2. Both manipulability maps are integrated, and the position of goods is determined.
3. Based on the position of goods, the base position of the robot is determined.

We set the robot as the origin of the robot coordinate system, assuming the frontal direction as the x -axis and the lateral direction as the y -axis. At each position on the XY plane, the manipulability is mapped for the situation in which objects are being carried by hand, as shown in Fig. 15a. This mapping is superimposed along the z -axis, which is the height direction, as shown in Fig. 15b. Thus, we create a three-dimensional manipulability map relative to the coordinate system of the robot. Similarly, we are also able to create a manipulability map for persons in Fig. 15b. Note that the current system uses a human model with a fixed size (1700 mm). However, we have proposed an estimation technique of a body shape using a statistical shape model of human and camera images [56], and thus we can apply the proposed technique for individual height adaptively with this technique.

The next step is to determine, using the manipulability map, the position of the goods that are about to be delivered. As shown in Fig. 16a, we take the maximum manipulability value according to each height, and retain the XY coordinates of each local coordinate system. These coordinates represent the relationship between the base position and the positions of the hands. We apply the same process to the coordinates of persons, thus superimposing the manipulability maps for robots and people, as shown in Fig. 16b. In doing so, the z -axis values on the manipulability map can be compensated for by using the face of the target person as a reference and synthesizing this data to suit the corresponding conditions of the person, for example, if the person is standing or sitting. As a result, the height value z in the absolute coordinate system used when delivering goods corresponds to the height of the sum of the maximum values of the manipulability.

According to the calculated height on the manipulability map for a person, the system requests the absolute coordinates of the goods to be delivered, using the previously retained relative coordinates of the hands. The position of the person that will receive the delivered goods is managed through TMS_SS and TMS_DB, and it is also possible to use this position as a reference point to request the position of the goods by fitting the relative coordinates. According to the aforementioned procedure, we can determine the unique position of the goods that are about to be delivered. As the final step, the base position of the robot is determined in order to hold out the goods to their previously calculated position. According to the manipulability map that corresponds to the height of a specific object, the system retrieves the relationship between the positions of hands and the base position. Using the position of the object as a reference point, the robot is able to hold the object out to any determined position if the base position meets the criteria of this relationship. Consequently,

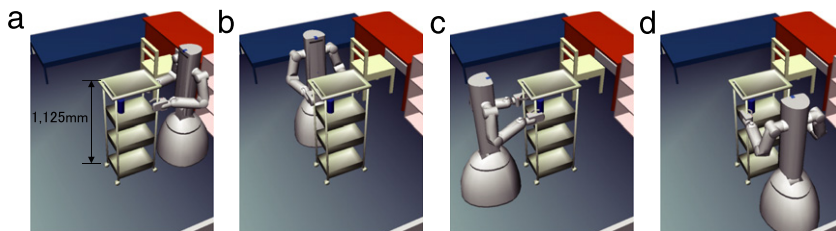


Fig. 14. Candidate points for wagon gripping position: (a) $i = 0$, (b) $i = 1$, (c) $i = 2$, (d) $i = 3$.

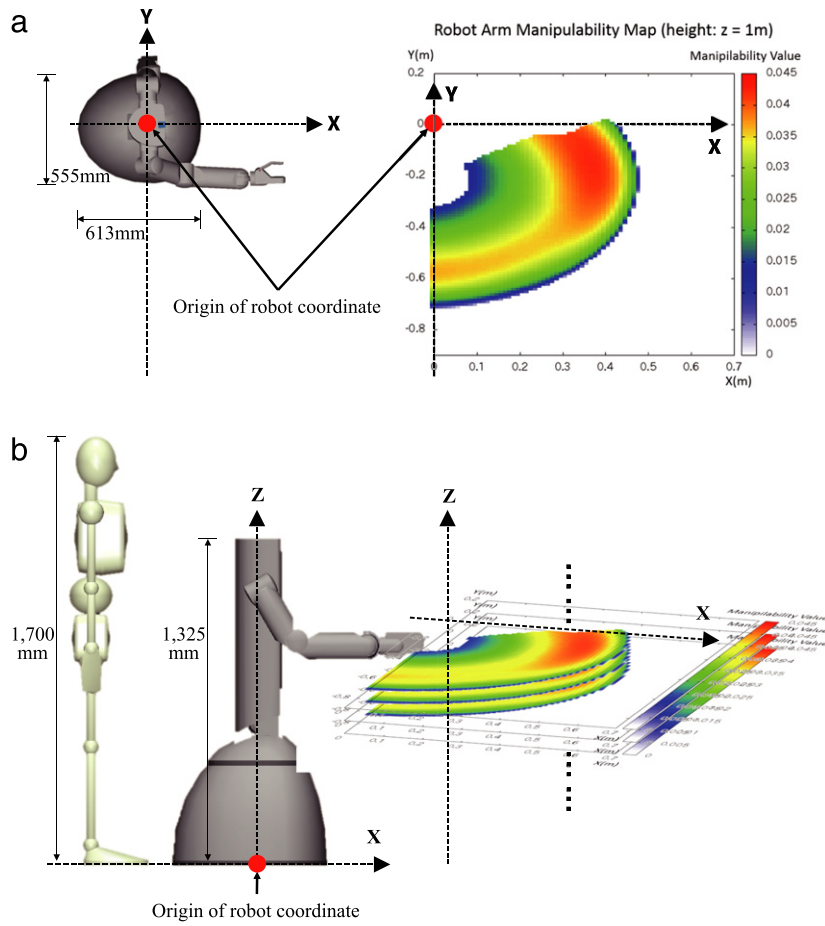


Fig. 15. Manipulability map for the robot: (a) 2D manipulability map for robot, (b) 3D manipulability map for robot.

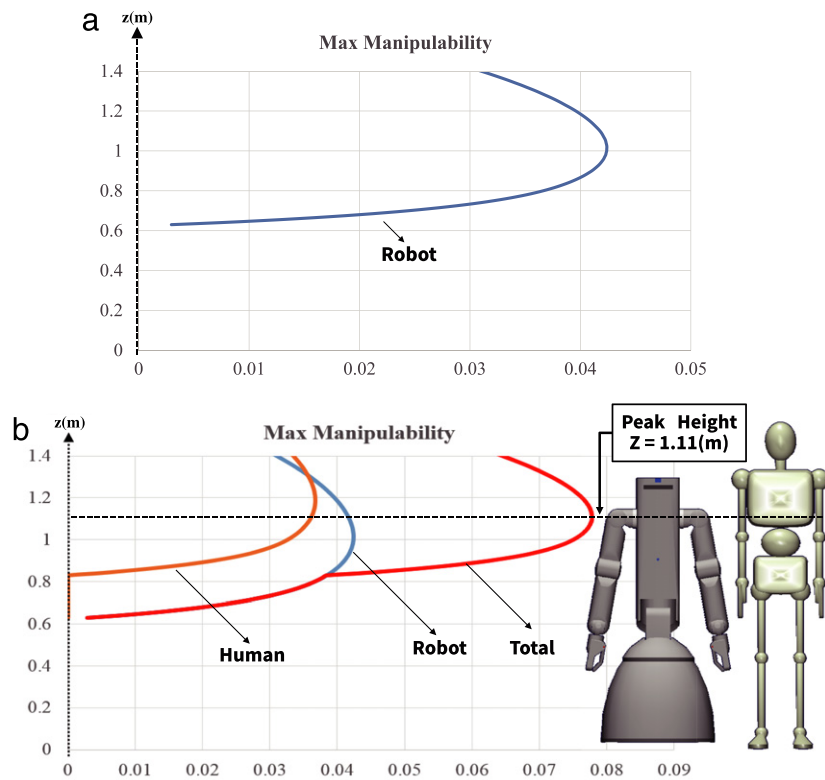


Fig. 16. Maximum value of manipulability: (a) maximum value of robot manipulability, (b) merge results for manipulability.

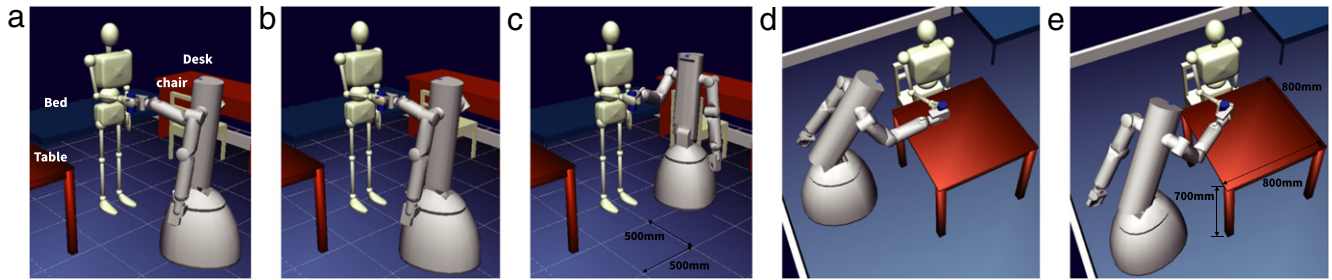


Fig. 17. Goods delivery position candidates.

Table 2
Evaluation of candidate goods delivery positions.

Position	a	b	c	d	e
View	1.00	1.00	0.00	0.00	1.00
D_{human}	1.08	1.14	0.78	0.81	0.97
D_{obs}	1.00	0.85	0.52	0.44	0.50
E_{value}	3.08	2.99	1.30	1.25	2.47

at the time of delivery, points on the circumference of the position of the object are determined to be candidate points on the absolute coordinate system of the base position. Considering all of the prospect points of the circumference, the following action planning, for which the system extracts multiple candidate points, is redundant. The best approach is to split the circumference n time, fetch a representative point out of each sector after the split, and limit the number of candidate points. After that, the obtained representative points are evaluated as in Eq. (7), while placing special emphasis on safety.

$$E_{give_obj_pos} = View + D_{human} + D_{obs} \quad (7)$$

Here, $View$ is a Boolean value that represents whether the robot enters the field of vision of the target person. If it is inside the field of vision, then $View$ is 1, otherwise $View$ is 0. This calculation is necessary because if the robot can enter the field of vision of the target person, then the robot can be operated more easily and the risk of unexpected contact with the robot is also reduced. In the above equation, D_{human} represents the distance to the target person, and D_{obs} represents the distance to the nearest obstacle. In order to reduce the risk of contact with the target person or an obstacle, the positions that represent the largest distance to the target person or obstacles are valued higher.

If all the candidate points on a given circumference sector result in contact with an obstacle, then the representative points of that sector are not selected. According to the aforementioned process, the base position of the robot is planned based on the position of the requested goods. The results for the case in which a person is standing still are shown in Fig. 17a through 17c, and the results for the case in which a person is sitting by a table are shown for Fig. 17d and e. Although the robot stands in front of a person in Fig. 17a and b, Fig. 17b is slightly closer to the object (a table). The corresponding evaluation results are shown in Table 2.

5.3. Movement path planning

5.3.1. Path planning for robots

Path planning for robots that serve in a general living environment requires a high degree of safety, which can be achieved by lowering the probability of contact with persons. However, for robots that push wagons, the parameter space that uniquely defines this state has a maximum of six dimensions, that is, position (x, y) and posture (θ) of a robot and a wagon, and planning a path that represents the highest safety values in such a

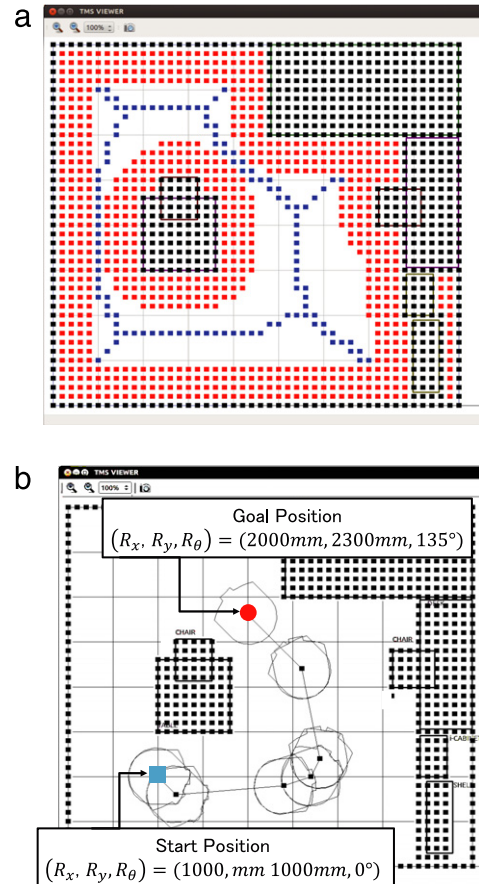


Fig. 18. Voronoi map and movement path: (a) collision area (red) and basic path (blue), (b) robot movement path.

space is time consuming. Thus, we require a method that produces a trajectory with a high degree of safety, but at the same time requires a short processing time. As such, we use a Voronoi map, as shown in Fig. 18.

5.3.2. Path planning for wagons

In order to be able to plan high-safety trajectories for wagons in real time, we need to reduce the dimensions of the path search space. The parameters that uniquely describe the state of a wagon pushing robot can have a maximum of six dimensions, but in reality the range in which the robot can operate the wagon is more limited. As such, for the case in which a robot is pushing a wagon, we set up a control point, as shown in Fig. 19, which fixes the relative positional relationship of the robot with the control point. The operation of the robot is assumed to change in terms of the relative orientation (W_θ) of the wagon with respect to the robot.

In addition, the range of relative positions is also limited. Accordingly, wagon-pushing robots are presented in just four

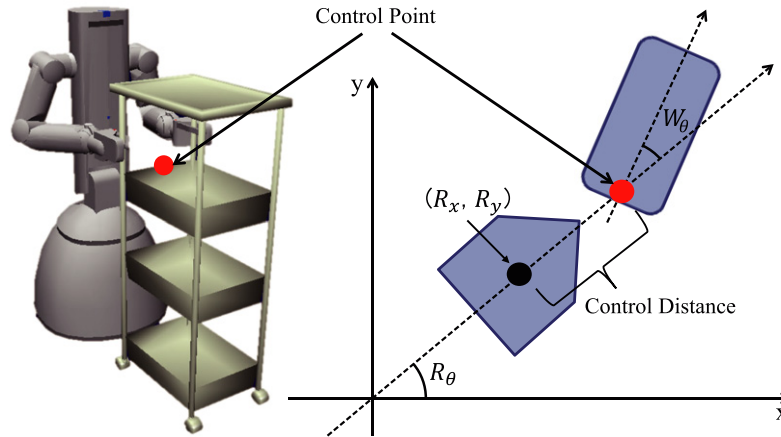


Fig. 19. Model of the robot pushing the wagon.

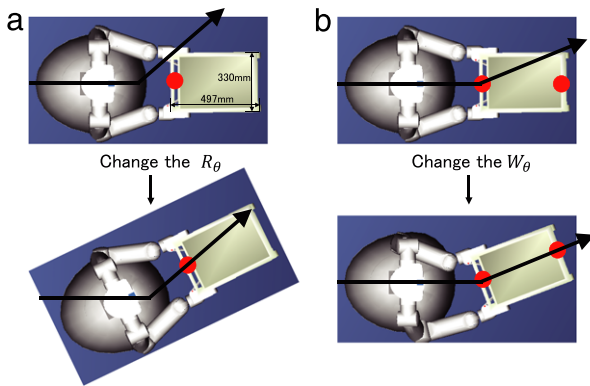


Fig. 20. Wagon path planning: (a) robot posture plan, (b) wagon posture plan.

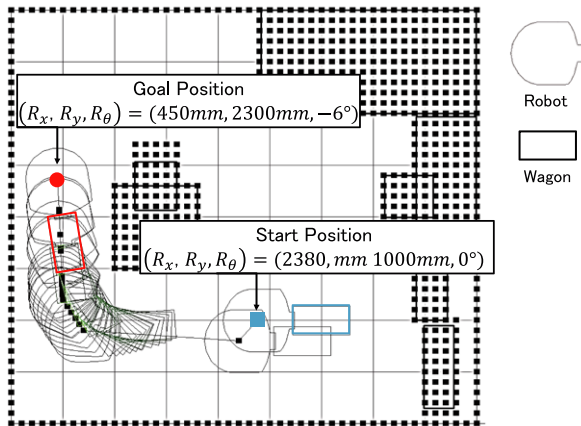


Fig. 21. Wagon movement path.

dimensions, which shortens the search time for the wagon path planning. Path planning for wagon-pushing robots uses the above-mentioned basic path and is executed as follows:

1. The start and end points are established.
2. The path for each robot along the basic path is planned.
3. According to each point on the path estimated in step 2, the position of the wagon control point is determined considering the manner in which the position of the wagon control point fits the relationship with the robot position.
4. If the wagon control point is not on the basic path (Fig. 20a), posture (R_θ) of the robot is changed so that the wagon control point passes along the basic path.

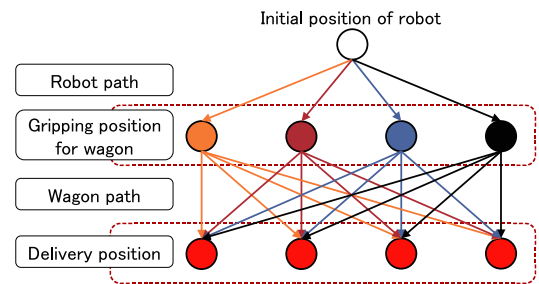


Fig. 22. Motion planning candidate.

5. If the head of the wagon is not on the basic path (Fig. 20b), the relative posture (W_θ) of the wagon is modified so that it passes along the basic path.
6. Steps 3 through 5 are repeated until the end point is reached.

Fig. 21 shows the results of wagon path planning, using example start and end points. These start $(R_x, R_y, R_\theta) = (2380 \text{ mm}, 1000 \text{ mm}, 0^\circ)$ and end points $(R_x, R_y, R_\theta) = (450 \text{ mm}, 2300 \text{ mm}, -6^\circ)$ use the outcomes of wagon grip position planning and position planning for goods delivery. This confirms that the movement traces of the wagon (indicated by green rectangles) are within the movement traces of the robot (indicated by the rounded gray shapes). Using this procedure we can simplify the space search without sacrificing the safety of the basic path diagram. The actual time required to calculate the path of a single robot was 1.10 (ms), and the time including the wagon path planning was 6.41 (ms).

5.4. Integration of planning

We perform operation planning for overall item-carrying action, which integrates position, path and arm motion planning. First, we perform wagon grip position planning in order for the robot to grasp a wagon loaded with goods. Next, we perform position planning for goods delivery in order to hand-deliver goods to the target person. The results of these work position planning tasks becomes the candidate movement target positions for the path planning of the robot and the wagon. Finally, we perform an action planning that combines the above-mentioned planning tasks, from the initial position of the robot to the path the robot takes until grasping the wagon, and the path the wagon takes until the robot reaches the position at which the robot can deliver the goods. For example, if there are four candidate positions for wagon gripping and four candidate positions for goods delivery around the target person, then we can plan 16 different actions, as shown in Fig. 22. The various action sequences obtained from this procedure are then evaluated to choose the optimum sequence.

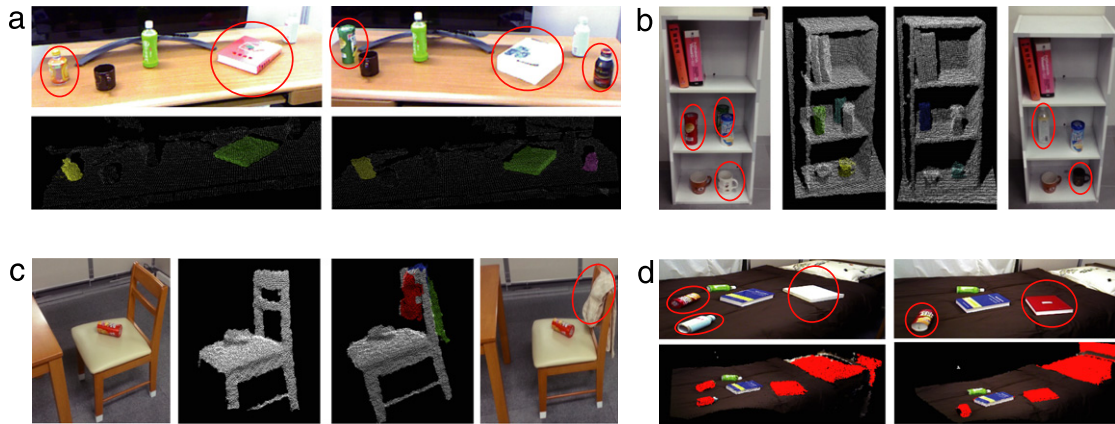


Fig. 23. Example of change detection: (a) table, (b) shelf, (c) chair, (d) bed.

Table 3
Experimental results.

	Change location	Detection	Detection rate [%]	Over detection
Table A	67	63	94.0	2
Table B	53	51	96.2	3
Total of Tables A and B	120	114	95.0	5
Shelf A	54	52	96.3	3
Shelf B	82	76	92.7	7
Total of shelves A and B	136	128	94.1	10
Chair	26	22	84.6	6
Bed	46	42	91.3	10

5.5. Evaluation of efficiency and safety

We evaluate each candidate action sequence based on efficiency and safety, as shown in Eq. (8).

$$E_{value} = \alpha \frac{Len_{min}}{Length} + \beta \frac{Rot_{min}}{Rotation} + \gamma ViewRatio \quad (8)$$

The α, β, γ are respectively the weight values of *Length*, *Rotation* and *ViewRatio*. The *Length* and *Rotation* represent the total distance traveled and total rotation angle. The Len_{min} and Rot_{min} represent the minimum values of all the candidate action. First and second terms of Eq. (8) are the metrics for efficiency of action. *ViewRatio* is the number of motion planning points in the person's visual field out of total number of motion planning point. It means the percentage for the number of motion planning points that exist in the visual field of person.

6. Experiments

In this section, we present the results of fundamental experiments described below using an actual robot and the proposed ROS-TMS.

1. Experiment to detect changes in the environment
2. Experiment to examine gripping and delivery of goods
3. Simulation of robot motion planning
4. Service experiments
5. Verification of modularity and scalability

6.1. Experiment to detect changes in the environment

We conducted experiments to detect changes using ODS (Section 4.3) with various pieces of furniture. We consider six pieces of target furniture, including two tables, two shelves, one chair, and one bed.

For each piece of furniture, we prepared 10 sets of previously stored data and newly acquired data of kinds of goods including

books, snacks, cups, etc., and performed point change detection separately for each set.

As the evaluation method, we considered the ratio of change detection with respect to the number of objects that were changed (change detection ratio). Furthermore, we also considered over-detection, which occurs when the system detects a change that has actually not occurred. The results of this experiment are shown in Table 3. The change detection ratios for each furniture type are as follows: 93.3% for tables, 93.4% for shelves, 84.6% for chairs, and 91.3% for beds. Examples of change detection for each piece of furniture are shown in Fig. 23. The sections enclosed by circles in each image represent points that actually underwent changes.

6.2. Experiment to examine gripping and delivery of goods

We performed an operation experiment in which a robot grasps an object located on a wagon and delivers the object to a person. As a prerequisite for this service, the goods are assumed to have been placed on the wagon, and their positions are known in advance. After performing the experiment 10 times, the robot successfully grabbed and delivered the object in all cases. The operating state is shown in Fig. 24.

We measured the displacement of the position of goods (O_x or O_y in Fig. 25) and the linear distance (d) between the measured value and the true value at the time of delivery, to verify the effect of rotation errors or arm posture errors. The results are listed in Table 4. Note that the truth value is measured using a tape measure.

The distance error of the position of the goods at the time of delivery was 35.8 mm. According to the manipulability degree, it is possible to cope with these errors, because the system plans a delivery posture with some extra margin in which persons and robots can move their hands.

6.3. Simulation of robot motion planning

We set up one initial position for the robot (R_x, R_y, R_θ) = (1000 mm, 1000 mm, 0°), the wagon (W_x, W_y, W_θ) = (3000 mm,



Fig. 24. Goods gripping experiment: (a) initial state, (b) intermediate position, (c) goods delivery attitude.

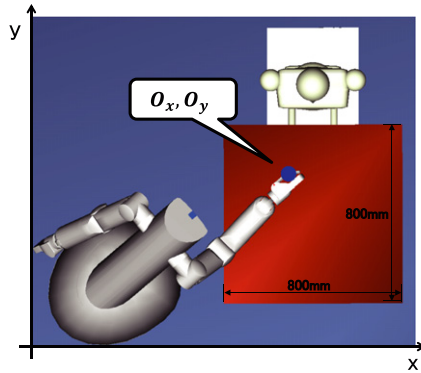


Fig. 25. Condition under which the goods delivery experiment was conducted.

Table 4
Results for goods delivery position.

	O_x (mm)	O_y (mm)	d (mm)
True value	1290.0	2100.0	0.0
Avg. of measured values	1301.9	2078.1	35.8
Avg. error	15.9	31.5	35.8
Maximum error	33.0	66.0	70.6
Minimal error	0.0	1.0	2.2

1000 mm, 0°), and the target person (H_x, H_y, H_θ) = (1400 mm, 2500 mm, -90°) and assume the person is in a sitting state as shown in Fig. 26a. Moreover, the range of vision of this person is shown in Fig. 26b by the red area. For each motion planning, two positions are candidate wagon grip positions (Fig. 27b and 28b), and three positions are candidate goods delivery positions (Fig. 27c–e and Fig. 28c–e), for a total of six possible action planning scenarios (Section 5.4).

The action planning result that passes over wagon grip candidate 1 is shown in Fig. 27, whereas the action planning result that passes over wagon grip candidate 2 is shown in Fig. 28. Furthermore, the evaluation values that changed the weight of each evaluation for each planning result are listed in Tables 5–7. The actions of Plan 2–3 were the most highly evaluated (Table 5). Fig. 28a and d indicate that all of the actions occur within the field of vision of the person. Since the target person can monitor the robot’s actions at all times, the risk of the robot unexpectedly touching a person is lower, and if the robot misses an action, the situation can be dealt with immediately. The action plan chosen from the above results according to the proposed evaluation values exhibits both efficiency and high safety.

6.4. Service experiments

We performed a service experiment for the carriage of goods, in accordance with the combined results of these planning sequences. The state of the sequence of actions is shown in Fig. 29.

The initial conditions were the same for the simulation (Fig. 26), and were set up as shown in Fig. 29a. The procedure for service execution is as follows:

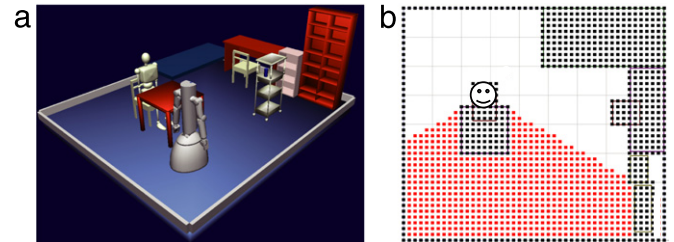


Fig. 26. Simulation conditions: (a) initial state of human, robot and wagon, (b) visual field of person.

Table 5
Evaluation results ($\alpha = \beta = \gamma = 1.0$).

Plan No.	1–1	1–2	1–3	2–1	2–2	2–3
Length	0.8	0.5	0.5	1.0	0.7	0.9
Rotation	0.6	0.4	0.4	0.9	0.9	1.0
View	0.5	0.3	0.8	0.8	0.9	1.0
Total Estimation	1.9	1.3	1.8	2.8	2.6	2.9

Table 6
Results of safety evaluation ($\alpha = \beta = 0.0, \gamma = 1.0$).

Plan No.	1–1	1–2	1–3	2–1	2–2	2–3
View Evaluation	0.5	0.3	0.8	0.8	0.9	1.0

1. Path planning is executed from the initial position to the position at which the object is delivered.
2. The robot moves from its initial position to the position at which it grips the wagon.
3. The robot position is corrected according to its RGB-D camera.
4. The arm trajectory is planned and executed in order to grasp the wagon (Fig. 29b).
5. The robot moves to the delivery position while pushing the wagon (Fig. 29c).
6. The arm trajectory is planned and executed in order to separate the arm from the wagon.
7. The arm trajectory is planned and executed in order to grasp the target object from the top of the wagon (Fig. 29d).
8. The arm trajectory is planned and executed in order to hold the object in the position in which it is delivered (Fig. 29e and f).

This service was carried out successfully, avoiding any contact with the environment. The total time for the task execution is 312 sec in case the maximum velocity of SmartPal-V is limited to 10 mm/sec in terms of safety. Moreover, the robot position was confirmed to always be within the range of vision of the subject during execution. Accordingly, we can say that the planned actions had an appropriate level of safety. Moreover, there was a margin for the movement of hands, as shown in Fig. 29f, for which the delivery process could appropriately cope with the movement errors of the robot.

Note that pushing and pulling motions of the wagon by the robot cause a large tracking error. In reality, the maximum error from desired trajectory was about 0.092 m in the experiments.

Table 7
Results of efficiency evaluation ($\alpha = \beta = 1.0, \gamma = 0.0$).

Plan No.	1-1	1-2	1-3	2-1	2-2	2-3
Length (mm)	4298.2	7336.9	6650.9	3443.5	4737.6	4045.6
Estimated length	0.8	0.5	0.5	1.0	0.7	0.9
Rotation (°)	362.3	478.8	480.6	225.5	233.6	212.6
Estimated rotation	0.6	0.4	0.4	0.9	0.9	1.0
Estimated efficiency	1.4	0.9	1.0	1.9	1.6	1.9

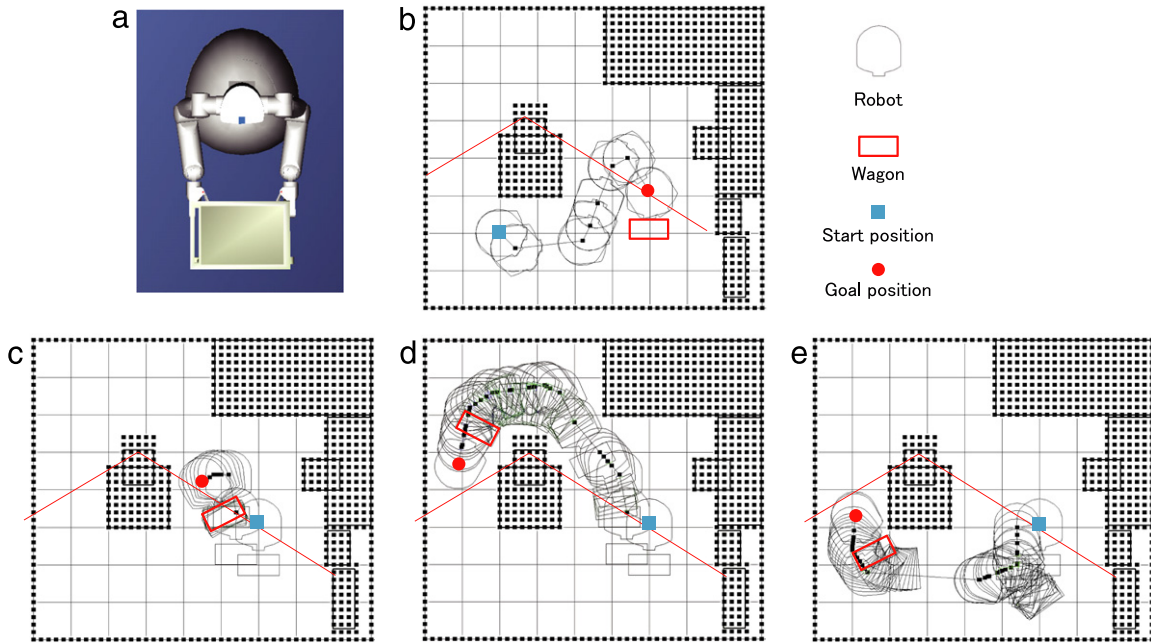


Fig. 27. Planning results for goods delivery wagon route for holding candidate position using plan 1 : (a) holding candidate attitude, (b) robot path to wagon grip candidate 1, (c, d, e) wagon route to goods delivery position candidate 1, 2, 3.

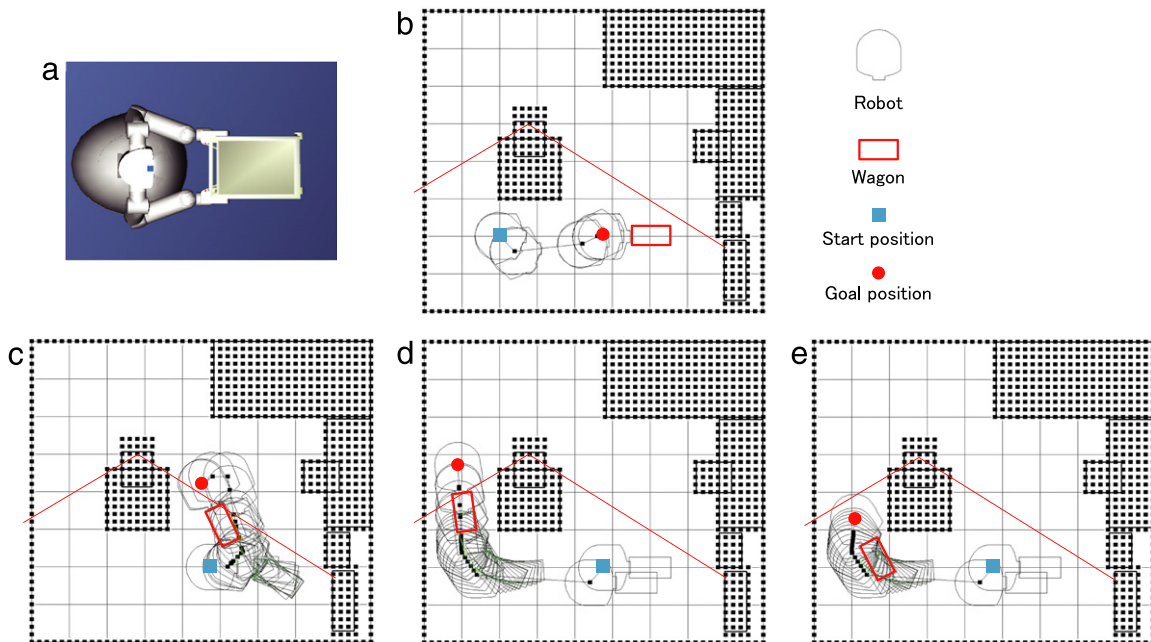


Fig. 28. Planning results for goods delivery wagon route for holding candidate position using plan 2 : (a) holding candidate attitude, (b) robot path to wagon grip candidate 2, (c, d, e) wagon route to goods delivery position candidate 1, 2, 3.

In worst-cases scenario, if the robot grasps the wagon rigidly and collides with obstacles, hands or wagon may be broken due to large impact force. In the experiments, we wrap a sponge around a

pole of the wagon to absorb the impact force. In addition, the start and goal positions are precisely measured by the optical poisoning system (Vicon MX) or the floor sensing system.

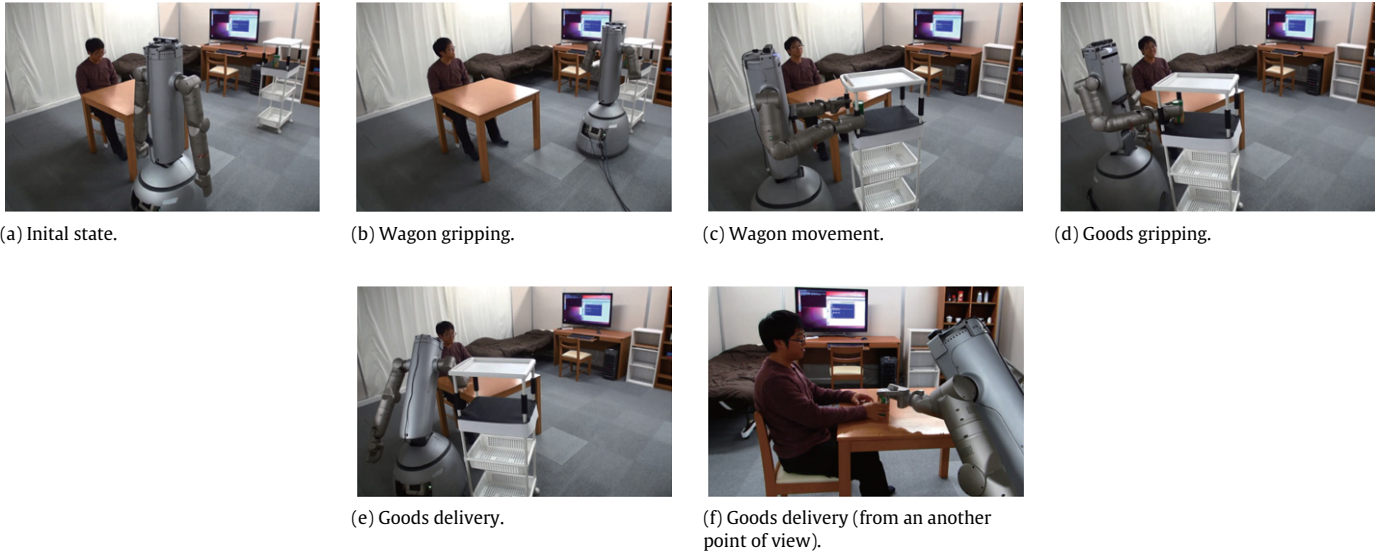


Fig. 29. State of the service execution.

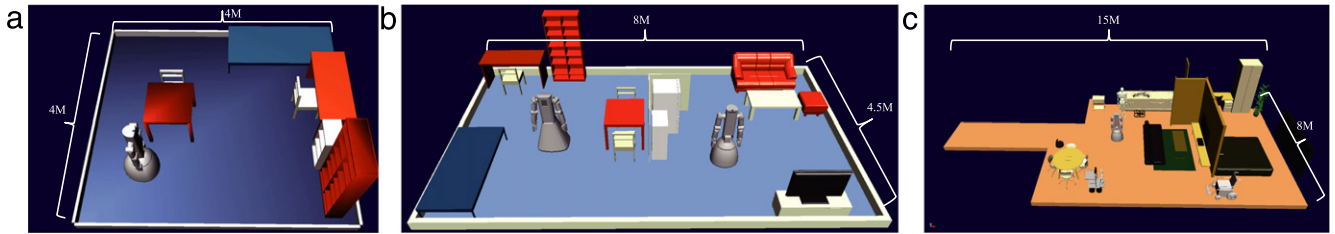


Fig. 30. Various environments managed by ROS-TMS: (a) Room A (4 m × 4 m, 18 packages, 32 processing nodes), (b) Room B (8 m × 4.5 m, 52 packages, 93 processing nodes), (c) Room C (15 m × 8 m, 73 packages, 151 processing nodes).

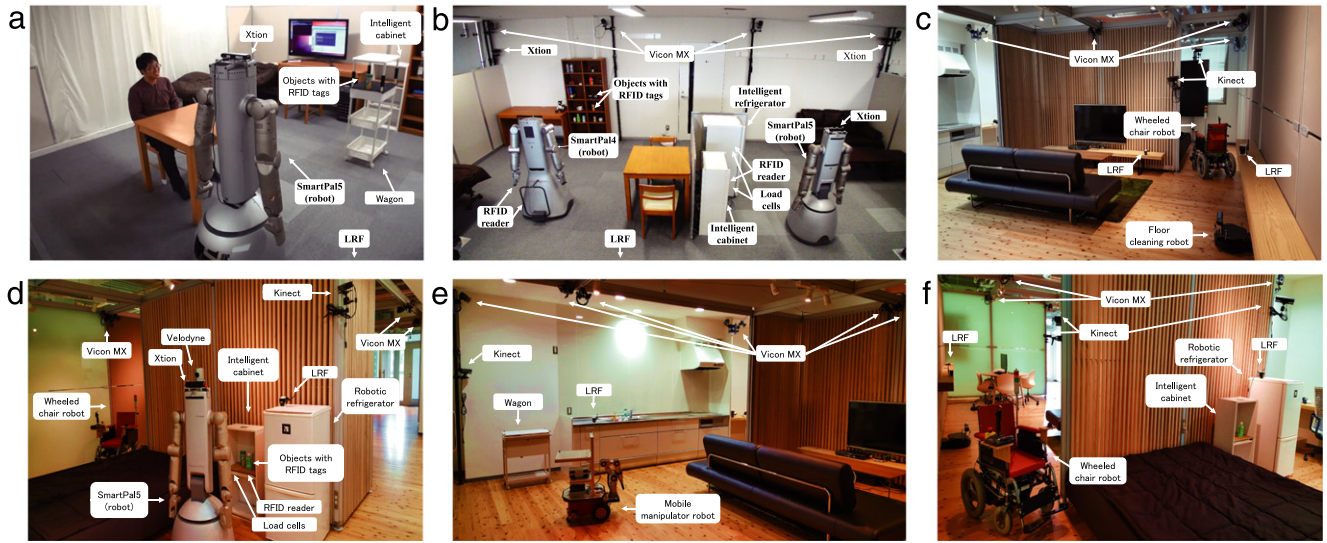


Fig. 31. Photos of real environments: (a) Room A, (b) Room B, (c)–(f) Room C.

6.5. Verification of modularity and scalability

We built the ROS-TMS for three types of rooms to verify its high modularity and scalability. The size and the structure of the rooms, sensor configurations, and variety of sensors and robots are all different between these rooms. The simulation models and actual photos of the rooms are shown in Figs. 30 and 31, respectively. In Room A (4 m × 4 m, Figs. 30a and 31a), we used 18 packages and 32 processing nodes including a LRF, three RFID tag readers, a

Xtion sensor and a humanoid robot. Room B (4.5 m × 8 m, Figs. 30b and 31b) consists of 52 packages and 93 processing nodes including two LRFs, two RFID tag readers, four Xtion sensors, ten Vicon cameras, two humanoid robots, a floor cleaning robot, a robotic refrigerator, and a wheeled chair robot. On the other hand, Room C (8 m × 15 m, Figs. 30c and 31c–f) utilizes 73 packages and 151 processing nodes including eight LRFs, two RFID tag readers, one Xtion sensor, five Kinect sensors, eighteen Vicon cameras, three humanoid robots, a floor cleaning robot, a mobile manipulator

robot, and a wheeled chair robot. Thanks to high flexibility and scalability of the ROS-TMS, we could set up these various environments in a comparatively short time.

7. Conclusions

In the present paper, we have introduced a service robot system with an informationally structured environment named ROS-TMS that is designed to support daily activities of elderly individuals. The room considered herein contains several sensors to monitor the environment and a person. The person is assisted by a humanoid robot that uses information about the environment to support various activities. In the present study, we concentrated on detection and fetch-and-give tasks, which we believe will be among most commonly requested tasks by the elderly in their daily lives. We have presented the various subsystems that are necessary for completing this task and have conducted several independent short-term experiments to demonstrate the suitability of these subsystems, such as a detection task using a sensing system and a fetch-and-give task using a robot motion planning system of the ROS-TMS. Currently, we adopt a deterministic approach for choosing proper data from redundant sensory information based on the reliability pre-defined manually. Our future work will include the extension to the probabilistic approach for fusing redundant sensory information. Also, we intend to design and prepare a long-term experiment in which we can test the complete system for a longer period of time.

Acknowledgments

This research is supported in part by a JSPS KAKENHI Grant Number 263244, a Grant-in-Aid for Scientific Research (A) (26249029) and the Japan Science and Technology Agency (JST) through its “Center of Innovation Science and Technology based Radical Innovation and Entrepreneurship Program (COI Program).”

We would like to thank all members of the ROS community and ROS development teams at Open Source Robotics Foundation.

References

- [1] K. Kogure, N. Hagita, Y. Sumi, N. Kuwahara, H. Ishiguro, Toward ubiquitous intelligent robotics, in: *Intelligent Robots and Systems, 2003. IROS 2003. Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, IEEE, 2003, pp. 1826–1831.
- [2] T. Hasegawa, K. Murakami, R. Kurazume, Y. Senta, Y. Kimuro, T. Ienaga, Robot town project: Sensory data management and interaction with robot of intelligent environment for daily life, in: *Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence, Pohang, Korea, 2007*, pp. 22–24.
- [3] S. Nishio, N. Hagita, T. Miyashita, T. Kanda, N. Mitsunaga, M. Shiomi, T. Yamazaki, Sensor network for structuring people and environmental information, *Cutting Edge Robot.* (2010) 367–378.
- [4] H. Noguchi, T. Mori, T. Sato, Automatic generation and connection of program components based on rdf sensor description in network middleware, in: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE, 2006, pp. 2008–2014.
- [5] Y. Kato, T. Izui, Y. Tsuchiya, M. Narita, M. Ueki, Y. Murakawa, K. Okabayashi, Rsi-cloud for integrating robot services with internet services, in: *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society, IEEE, 2011*, pp. 2158–2163.
- [6] H. Gross, C. Schröter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, T. Langner, C. Martin, M. Merten, I'll keep an eye on you: Home robot companion for elderly people with cognitive impairment, in: *Systems, Man, and Cybernetics, SMC, 2011 IEEE International Conference on*, IEEE, 2011, pp. 2481–2488.
- [7] C.D. Nugent, D.D. Finlay, P. Fiorini, Y. Tsumaki, E. Prassler, Editorial home automation as a means of independent living, *IEEE Trans. Autom. Sci. Eng.* 5 (1) (2008) 1–9.
- [8] K. Kayama, I.E. Yairi, S. Igi, Semi-autonomous outdoor mobility support system for elderly and disabled people, in: *Intelligent Robots and Systems, 2003. IROS 2003. Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, IEEE, 2003, pp. 2606–2611.
- [9] T. Mori, A. Takada, H. Noguchi, T. Harada, T. Sato, Behavior prediction based on daily-life record database in distributed sensing space, in: *Intelligent Robots and Systems, 2005. IROS 2005. 2005 IEEE/RSJ International Conference on*, IEEE, 2005, pp. 1703–1709.
- [10] Y. Nakauchi, K. Noguchi, P. Somwong, T. Matsubara, A. Namatame, Vivid room: human intention detection and activity support environment for ubiquitous autonomy, in: *Intelligent Robots and Systems, 2003. IROS 2003. Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, IEEE, 2003, pp. 773–778.
- [11] Y. Nishida, H. Aizawa, T. Hori, N.H. Hoffman, T. Kanade, M. Kakikura, 3d ultrasonic tagging system for observing human activity, in: *Intelligent Robots and Systems, 2003. IROS 2003. Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, IEEE, 2003, pp. 785–791.
- [12] K. Murakami, K. Matsuo, T. Hasegawa, R. Kurazume, Position tracking and recognition of everyday objects by using sensors embedded in an environment and mounted on mobile robots, in: *Robotics and Automation, ICRA, 2012 IEEE International Conference on*, IEEE, 2012, pp. 2210–2216.
- [13] K. Kawamura, M. Iskarous, Trends in service robots for the disabled and the elderly, in: *Intelligent Robots and Systems' 94. 'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 3, IEEE, 1994, pp. 1647–1654.
- [14] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, et al. Towards personal service robots for the elderly, in: *Workshop on Interactive Robots and Entertainment, WIRE 2000*, vol. 25, 2000, page 184.
- [15] M. Kim, S. Kim, S. Park, M.-T. Choi, M. Kim, H. Goma, Service robot for the elderly, *IEEE Robot. Autom. Mag.* 16 (1) (2009) 34–45.
- [16] T. Hasegawa, K. Murakami, Robot town project: Supporting robots in an environment with its structured information, in: *Proc. of The 3rd International Conference on Ubiquitous Robots and Ambient Intelligence, 2006*, pp. 119–123.
- [17] K. Murakami, T. Hasegawa, R. Kurazume, Y. Kimuro, A structured environment with sensor networks for intelligent robots, in: *Sensors, 2008 IEEE, IEEE, 2008*, pp. 705–708.
- [18] B.K. Kim, N. Tomokuni, K. Ohara, T. Tanikawa, K. Ohba, S. Hirai, Ubiquitous localization and mapping for robots with ambient intelligence, in: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE, 2006, pp. 4809–4814.
- [19] R. Kurazume, Y. Iwashita, K. Murakami, T. Hasegawa, Robot town project and 3-d co-operative geometrical modeling using multiple robots, in: *15th International Symposium on Robotics Research, ISRR, International Foundation for Robotics Research (IFRR), 2011*.
- [20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, Ros: an open-source robot operating system, *ICRA Workshop on Open Source Software 3* (3.2) (2009) 5.
- [21] M. Inoue, K. Uemura, Y. Minagawa, M. Esaki, Y. Honda, A home automation system, *Consumer Electronics, IEEE Trans. Consum. Electron.* 31 (3) (1985) 516–527.
- [22] A. Alharaish, Design and implementation of home automation system, *IEEE Trans. Consum. Electron.* 50 (4) (2004) 1087–1092.
- [23] K. Gill, S.-H. Yang, F. Yao, X. Lu, A zigbee-based home automation system, *IEEE Trans. Consum. Electron.* 55 (2) (2009) 422–430.
- [24] J.-H. Lee, H. Hashimoto, Intelligent space concept and contents, *Adv. Robot.* 16 (3) (2002) 265–280.
- [25] J.-H. Lee, K. Morioka, N. Ando, H. Hashimoto, Cooperation of distributed intelligent sensors in intelligent environment, *IEEE/ASME Trans. Mechatronics* 9 (3) (2004) 535–543.
- [26] D. Brščić, H. Hashimoto, Map building and object tracking inside intelligent spaces using static and mobile sensors, in: *IEEE/RSJ International Conference on Robotic Systems, IROS 2007*.
- [27] K. Morioka, J.-H. Lee, H. Hashimoto, Intelligent space for human centered robotics, *Adv. Serv. Robot.* (2008) 181–192.
- [28] A. Saffiotti, M. Broxvall, Peis ecologies: Ambient intelligence meets autonomous robotics, in: *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies, ACM, 2005*, pp. 277–281.
- [29] M. Broxvall, B.-S. Seo, W. Kwon, The peis kernel: A middleware for ubiquitous robotics, in: *Proc. of the IROS-07 Workshop on Ubiquitous Robotic Space Design and Applications, 2007*, pp. 212–218.
- [30] M. Arndt, S. Wille, L. De Souza, V.F. Rey, N. Wehn, K. Berns, Performance evaluation of ambient services by combining robotic frameworks and a smart environment platform, *Robot. Auton. Syst.* 61 (11) (2013) 1173–1185.
- [31] J.-H. Kim, Y.-D. Kim, K.-H. Lee, The third generation of robotics: Ubiquitous robot, in: *Proc of the 2nd Int. Conf. on Autonomous Robots and Agents*.
- [32] J.-H. Kim, Ubiquitous robot, in: *Computational Intelligence, Theory and Applications, Springer, 2005*, pp. 451–459.
- [33] J.-H. Kim, K.-H. Lee, Y.-D. Kim, N.S. Kuppaswamy, J. Jo, Ubiquitous robot: A new paradigm for integrated services, in: *Robotics and Automation, 2007 IEEE International Conference on*, IEEE, 2007, pp. 2853–2858.
- [34] M. Waibel, M. Beetz, J. Civera, R. d'Andrea, J. Elfiring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, et al., A world wide web for robots, *IEEE Robot. Autom. Mag.*
- [35] M. Tenorth, A.C. Perzylo, R. Lafrenz, M. Beetz, The roboearth language: Representing and exchanging knowledge about actions, objects, and environments, in: *Robotics and Automation, ICRA, 2012 IEEE International Conference on*, IEEE, 2012, pp. 1284–1289.
- [36] N. Hubel, G. Mohanarajah, R. van de Molengraft, M. Waibel, R. D'Andrea, Roboearth project [online] (2015).

- [37] N.S.M. Nor, M. Mizukawa, Robotic services at home: An initialization system based on robots information and user preferences in unknown environments, *Int. J. Adv. Robot. Syst.* 11.
- [38] K. Ukai, Y. Ando, M. Mizukawa, Robot technology ontology targeting robot technology services in kukanchi—"interactive human-space design and intelligence", *J. Robot. Mechatronics* 21 (4) (2009) 489.
- [39] Ieee symposium on robotic intelligence in informationally structured space [online] (2015).
- [40] K.L. Koay, E.A. Sisbot, D.S. Syrdal, M.L. Walters, K. Dautenhahn, R. Alami, Exploratory study of a robot approaching a person in the context of handing over an object, in: AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics, 2007, pp. 18–24.
- [41] A. Edsinger, C.C. Kemp, Human-robot interaction for cooperative manipulation: Handing objects to one another, in: *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on, IEEE, 2007*, pp. 1167–1172.
- [42] F. Dehais, E.A. Sisbot, R. Alami, M. Causse, Physiological and subjective evaluation of a human-robot object hand-over task, *Appl. Ergon.* 42 (6) (2011) 785–791.
- [43] K.W. Strabala, M.K. Lee, A.D. Dragan, J.L. Forlizzi, S. Srinivasa, M. Cakmak, V. Micelli, Towards seamless human-robot handovers, *J. Human-Robot Interact.* 2 (1) (2013) 112–132.
- [44] J. Tan, N. Xi, Integrated sensing and control of mobile manipulators, in: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 2, IEEE, 2001, pp. 865–870.
- [45] J. Tan, N. Xi, Y. Wang, Integrated task planning and control for mobile manipulators, *Int. J. Robot. Res.* 22 (5) (2003) 337–354. <http://ijr.sagepub.com/content/22/5/337.full.pdf+html> <http://dx.doi.org/10.1177/0278364903022005004>.
- [46] F. Lamiroux, D. Bonnafous, O. Lefebvre, Reactive path deformation for nonholonomic mobile robots, *IEEE Trans. Robot.* 20 (6) (2004) 967–977.
- [47] M. Stilman, J.J. Kuffner, Navigation among movable obstacles: Real-time reasoning in complex environments, *Int. J. Humanoid Robot.* 2 (04) (2005) 479–503.
- [48] S. Nozawa, T. Maki, M. Kojima, S. Kanzaki, K. Okada, M. Inaba, Wheelchair support by a humanoid through integrating environment recognition, whole-body control and human-interface behind the user, in: *Intelligent Robots and Systems, 2008. IROS 2008, IEEE/RSJ International Conference on, IEEE, 2008*, pp. 1558–1563.
- [49] J. Scholz, S. Chitta, B. Marthi, M. Likhachev, Cart pushing with a mobile manipulation system: Towards navigation with moveable objects, in: *Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011*, pp. 6115–6120.
- [50] S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control, in: *Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012*, pp. 1428–1435.
- [51] S. Nakaoka, Choreonoid: Extensible virtual robot environment built on an integrated gui framework, in: *IEEE/SICE International Symposium on System Integration*.
- [52] Y. Nohara, T. Hasegawa, K. Murakami, Floor sensing system using laser range finder and mirror for localizing daily life commodities, in: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010*, pp. 1030–1035.
- [53] Y. Pyo, T. Hasegawa, T. Tsuji, R. Kurazume, K. Morooka, Floor sensing system using laser reflectivity for localizing everyday objects and robot, *Sensors* 14 (4) (2014) 7524–7540.
- [54] K. Murakami, T. Hasegawa, K. Shigematsu, F. Sueyasu, Y. Nohara, B.W. Ahn, R. Kurazume, Position tracking system of everyday objects in an everyday environment, in: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010*, pp. 3712–3718.
- [55] P. Alimi, D. Meger, J.J. Little, Object persistence in 3d for home robots, in: *Semantic Perception, Mapping and Exploration 7*.
- [56] M. Shinzaki, Y. Iwashita, K. Ogawara, R. Kurazume, Gait-based person identification method using shadow biometrics for robustness to changes in the walking direction, in: *IEEE Winter Conference on Applications of Computer Vision 2015 (WACV), IEEE, 2015*, pp. 670–677.



Yoonseok Pyo received his B.E. degree from the Electronics & Information Engineering, Kwangwoon University, in 2009. He worked as a research fellow of Korea Institute of Science and Technology (KIST) from 2009 to 2010. His M.S. degree was from the Graduate School of Information Science and Electrical Engineering, Kyushu University, in 2013. He is a research fellow of the Japan Society for the Promotion of Science. He is currently working on the extension of distributed sensor and service robot platform as a Ph.D. student of the Graduate School of Information Science and Electrical Engineering, Kyushu University.



Kouhei Nakashima received his B.E. degree from the Graduate School of Information Science and Electrical Engineering, Kyushu University, in 2012. He is currently a M.S. student of the Graduate School of Information Science and Electrical Engineering, Kyushu University. His research interests include motion planning and software platform of robotic systems.



Shunya Kuwahata received his B.E. degree from the Graduate School of Information Science and Electrical Engineering, Kyushu University, in 2012. He is currently working on the extension of vision application using RGB-D camera as a M.S. student of the Graduate School of Information Science and Electrical Engineering, Kyushu University.



Ryo Kurazume received his M.E. and B.E. degrees from the Department of Mechanical Engineering Science, Tokyo Institute of Technology, in 1991 and 1989, respectively. His Ph.D. degree was from the Department of Mechanical Engineering Science, Tokyo Institute of Technology, in 1998. He is a professor at the Graduate School of Information Science and Electrical Engineering, Kyushu University. His current research interests include multiple mobile robots, 3D modeling, manipulator, walking robots, and medical image analysis.



Tokuo Tsuji received his B.S., M.S., and Doctoral degrees from Kyushu University in 2000, 2002 and 2005, respectively. He worked as a Research Fellow of Graduate School of Engineering, Hiroshima University from 2005 to 2008. He worked as a Research Fellow of Intelligent Systems Research Institute of National Institute of Advanced Industrial Science and Technology (AIST) from 2008 to 2011. From 2011, he has been working as a Research Associate at Kyushu University. His research interests include multi-fingered hand, machine vision, and software platform of robotic systems. He is a member of IEEE, JSME, RSJ, and IE-

ICE.



Ken'ichi Morooka received his M.S. and Ph.D. degrees from the Graduate School of Information Science and Electrical Engineering, Kyushu University, in 1997 and 2000, respectively. He was associated with Institute of Systems & Information Technologies/KYUSHU. From 2000 to 2006, he was Research Associate at Imaging Science and Engineering Laboratory, Tokyo Institute of Technology. From 2006 to 2010, he was Associate Professor in Digital Medicine Initiative, Kyushu University, Japan. Currently, he is Associate Professor in Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan. His research interests cover medical imaging, computer graphics, and pattern recognition.

Japan. His research interests cover medical imaging, computer graphics, and pattern recognition.



Tsutomu Hasegawa received his B.E. degree in Electronic Engineering and his Ph.D. degree both from the Tokyo Institute of Technology, in 1973 and 1987, respectively. He was associated with the Electrotechnical Laboratory of the Japanese Government from 1973 to 1992, where he performed research in robotics. From 1981 to 1982, he was a Visiting Researcher at the Laboratoire d'Automatique et d'Analyses des Systemes (LAAS/CNRS), Toulouse, France. From 1992 to 2013, he was a Professor with the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University. He is currently Director of Kumamoto National College of Technology. His research interests are in manipulator control, geometric modeling and reasoning, motion planning, and man-machine interaction. He received the Franklin V. Taylor Memorial Award from the IEEE Systems, Man and Cybernetics Society in 1999, and the 2013 JSME Medal for Distinguished Engineers in 2014.

University. He is currently Director of Kumamoto National College of Technology. His research interests are in manipulator control, geometric modeling and reasoning, motion planning, and man-machine interaction. He received the Franklin V. Taylor Memorial Award from the IEEE Systems, Man and Cybernetics Society in 1999, and the 2013 JSME Medal for Distinguished Engineers in 2014.