Development of Robotic Intelligent Space Using Multiple RGB-D Cameras for Industrial Robots

Ryo KABUTAN, Ryodo TANAKA, Shinji OOMORI, Masaru MORITA, Eiichi INOHIRA, Kaori YOSHIDA, Hiroshi OHTAKE, Takeshi NISHIDA

Abstract— **To create industrial robots that can revise their own work by an autonomous decision, we propose a novel industrial robot sensor system called the "robotic intelligent space (RIS)". The system enables real-time measurement of the industrial robots and the objects manipulated in the working space by using multiple RGB-D cameras. The RGB-D cameras are installed in various places in the RIS to reduce occlusions, and information is merged in a three-dimensional occupancy grid map with real-time updates. Moreover, the RIS provides precise recognition of obstacles, and robot motion planning using the map. We developed a prototype of the RIS with two RGB-D cameras and an industrial robot. Additionally, we constructed an object-handling application including a position and posture recognition function. We demonstrate the efficiency and performance of the RIS through several simulations and experimental results.**

I. Introduction

Recently, many studies with the aim of sophistication of industrial robots have been carried out. These studies are not aimed towards autonomous robots, but rather involve coordination and collaboration with humans [1, 2]. In order to improve the teaching playback system, which is currently standard in factories, a more flexible and rapid system that can adapt to the multi-product variable production has been researched.

In this research, to create the entire intelligent space around the robot, we developed a novel sensor system for autonomous working industrial robots. This sensor system measures three-dimensional (3D) real-time information around the industrial vertical articulated robot, even if the targets are manipulated or if it is in the presence of a working person. The obtained measurement data are analyzed for reorganization of manipulation targets and all of the usable data sets are integrated into a simulator. All of them are visualized. The user can select and adjust the information of the visualized data and can thereby predict the near future of the robot system. We call this virtualized space the robotic intelligent space (RIS). Moreover, we developed a novel user interface for easy operation of the RIS. A humanoid robot is introduced as the control interface, and users can re-program the industrial robot in the RIS by talking to and touching the humanoid robot (Fig. 1 (a)). In this paper, the development and details of the RIS system are described, and its effectiveness is evaluated through several simulation results.

R. Kabutan, R. Tanaka, S. Oomori, M. Morita, E. Inohira, K. Yoshida, H. Ohotake, T. Nishida are with Kyushu Institute of Technology, Kitakyushu, Fukuoka, Japan (corresponding author's e-mail: nishida@cntl.kyutech.ac.jp).

II. Overview of the RIS

In this study, the RIS includes two Kinects. A Kinect is an inexpensive RGB-D camera that can obtain point cloud data. RIS uses Motoman SIA5 (Yaskawa Electric Corp.). In addition, we use D-Hand (made by DOUBLE Research and Development) as the robot end effector. The gripper is a three finger-shaped hand that can execute a gripping operation with

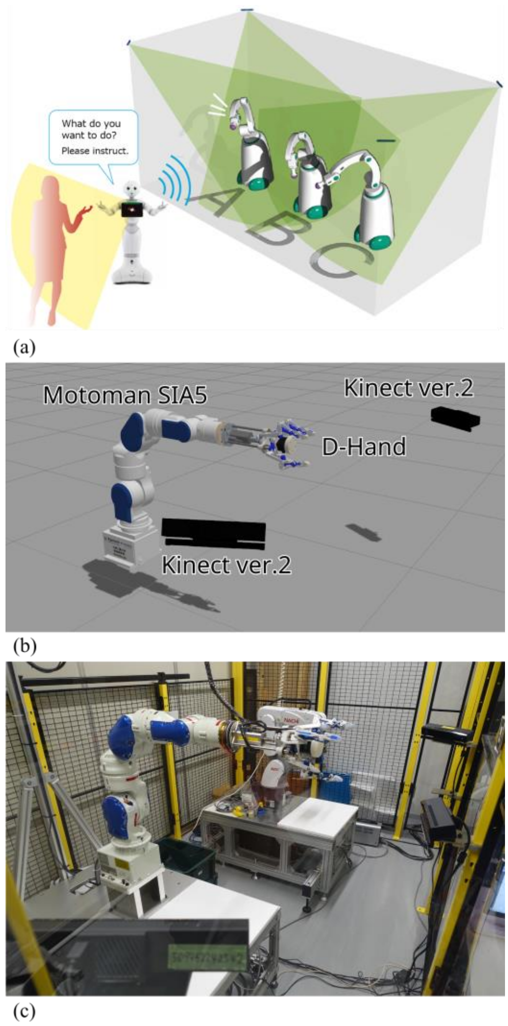


Fig. 1. Constitution of RIS: (a) we propose a next-generation industrial robot system, (b) overview of the physical simulator, and (c) the real experimental environment of RIS.

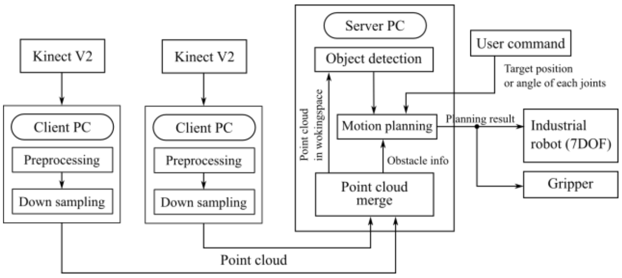
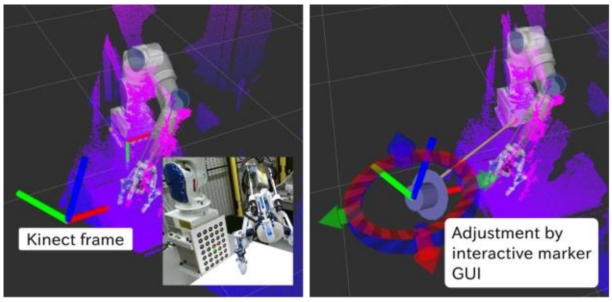


Fig. 2. Conceptual model.



(a) (b)

Fig. 3. Caribration of the camera external parameters: (a) calibration with a checkerboard, (b) manual calibration function in the simulator.

one actuator. The RIS system is developed by using these devices (Fig. 1 (b), (c)).

A. *Industrial robot*

The RIS system is intended to automatically move the industrial robot (SIA5) without requiring complicated robot teaching. This robot has seven degrees of freedom (7-DOF). Therefore, it can strictly avoid obstacles and can assume a more complicated posture compared to a 6-DOF manipulator.

The robot end effector (D-hand) follows the shape of a target object and can grasp it. Because of its unique mechanical structure, it has a high success rate in terms of grasping.

B. *Kinect for Windows ver.2*

We use Kinect for Windows ver.2 (Microsoft Corp.) which is a rather inexpensive sensor. Kinect ver.2 is a RGB-D camera for measuring the three-dimensional environment. On one hand, Kinect ver.1 adopts the active stereo method—which consists of projecting a specific pattern of infrared laser on an object, capturing the image in that state, and calculating the depth of each point on the image. Hence, using Kinect ver.1 does not allow the RIS to use multiple Kinects simultaneously. However, when we built the RIS, we chose Kinect ver.2, which adopts the Time-of-Flight (TOF) method. Kinects are placed in front of the robot (Fig. 1 (b), (c)). We made the space less occluded by merging the point clouds generated from each Kinect.

C. *ROS (robot operating system) [3]*

Our system runs on ROS Indigo and Ubuntu Trusty (14.04). Robot operation commands, motion planning, point cloud process, and similar operations are calculated by three PCs. All processing is programmed as ROS packages, carried out by distributed processing.

III. ELEMENTAL TECHNOLOGY TO BUILD RIS

In Section III, we present some research techniques (Fig. 2) in detail. Utilizing the RIS, the industrial robot autonomously performs obstacle avoidance and pick and place tasks. Fig. 2 shows a summary of the whole system.

A. *Kinect extrinsic calibration and initial point cloud processing*

We calibrate the extrinsic parameters (position and posture) of two Kinects with a checkerboard. Because the point clouds obtained from two Kinects are merged, we adjust the position parameters of the Kinects precisely. For this purpose, an initial rough setting position $(x, y, z, \mathrm{roll}, \mathrm{pitch}, \mathrm{yaw})$ of Kinect is estimated using the checkerboard (Fig. 3(a)). Following this, the Kinect's frame is precisely calibrated using interactive marker GUI (Fig. 3(b)). The point cloud obtained from one Kinect is pretreated by one client PC (Fig. 2). Briefly, pretreatments involve removing the noise from measurements and filtering within the specified area, that is, the robot working space. After the pretreated point cloud dataset is generated, the points are reduced (down sampling) considering the required density of the point cloud. Finally, on completion of the processing, the dataset is published to the server PC.

B. *Object detection*

The RIS is available for detecting objects, such as target objects placed on a table. Moreover, it enables the automatic execution of pick and place tasks for the industrial robot. In order to satisfy this, it is necessary to estimate the position and posture. Fig. 4 shows the flow of object detection and the creation of bounding boxes.

(a) Using the point cloud dataset obtained from the RIS, Euclidean clusters are extracted. The calculated cluster area is limited to the table where the robot is installed (Fig. 5 (a)).

(b) From the results, a three-dimensional point cloud dataset is projected on the two- dimensional x-y plane.

(c) We reduce the dimension of the point cloud to determine the convex hull.

(d) The minimum bounding rectangle for the convex hull is searched.

(e) We obtain the rotation and the center of gravity of the solved minimum bounding rectangle.

(f) After the bounding box in three-dimensional space is determined by the solution provided on the two-dimensional plane, this process provides the object

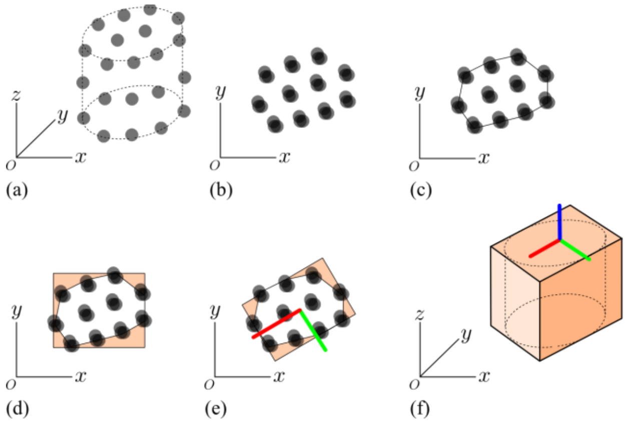


Fig. 4. Flow of object detection: (a) Clustering result. (b) Projecting 3d point to 2d point. (c) Finding the convex hull. (d) Searching minimum bounding rectangle. (e) Obtaining rotation of the minimum bounding. (f) Finally, returning to the center of mass and rotation.

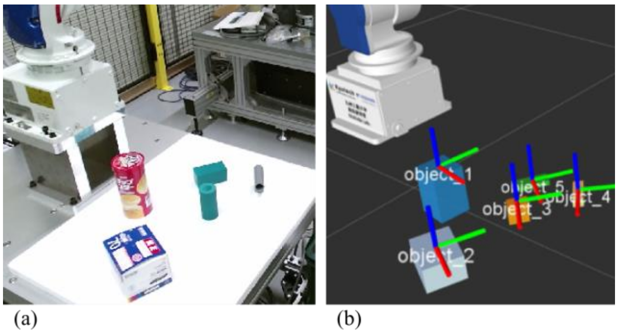


Fig. 5. Result of object detection: (a) placing objects in the workspace, and (b) object detection result.

information: length, width, height, center of gravity, and rotation (Fig. 5 (b))

The object detection algorithm is calculated by the server PC (Fig. 2).

C. *Motion planning*

We set the target position, which is provided by the perception algorithm (Subsection B), and we generate the motion for object grasping. In addition, the RIS enables the creation of the occupancy grid map around the robot. We use this obstacle information of the motion planning for obstacle avoidance. In this system, motion planning is performed using “MoveIt!”, a motion planning library based on ROS, which is also provided as a user interface for motion planning. First, we create a three-dimensional occupancy grid map for obstacle avoidance. This consists of the merged cloud dataset obtained from the RIS. Comparisons are made between two types of conditions: in the case of occupancy grid map made from a single Kinect and the RIS (Fig. 6). The single Kinect cannot assist in generating accurate robot motion, because there is an occlusion region behind the obstacle (Fig. 6 (b)). On the other hand, we install multiple Kinects to compensate for occlusions when using the RIS (Fig. 6 (c) and (d)). It is possible to correctly obtain the shape of the original obstacle.

Next, we create an application of the RIS to automatically pick up the target objects. In this system, the pick and place positions are changed randomly. Therefore, motion planning is carried out each time for pick and place tasks. Because we have to improve the user experience on pick and place tasks, a planning algorithm is required to ensure speed and efficiency. Accordingly, we use RRT-Connect, which has been evaluated especially in terms of search speed in the proposed motion planning algorithm [4]. Fig. 7 shows the result of generating a trajectory for executing pick and place tasks after detecting the working object. Moreover, Table I shows the result after repeating the experiment to execute pick and place tasks 120 times (as shown in Fig. 7). The objects are placed in random positions for each trial.

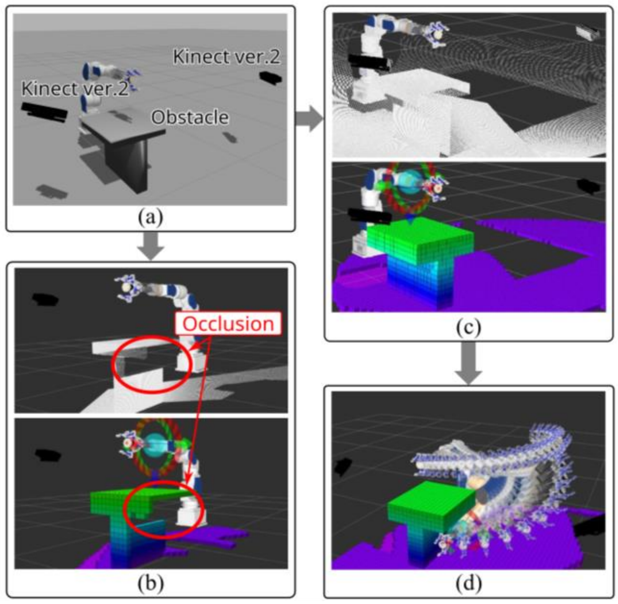


Fig. 6. Occupancy grid map for obstacle avoidance: (a) state of the robot and obstacle, (b) case of single Kinect, (c) obstacle avoidance with multiple Kinects, and (d) the RIS enables generation of precise motion planning.

D. *User interface for operating industrial robot using Pepper*

In this section, we present an example of a system using the RIS for an industrial robot that does not require a skilled operator. In "User command" (Fig. 2), we can select an object from the identified objects, and in which box the object has to be placed. The job command is carried out through Pepper (Softbank Robotics Corp.). This system enables operation of the industrial robot by abstract directive without operating the teaching pendant. Because Pepper has a voice guidance function and tablet display, it can send the operation command to the industrial robot (Fig. 8). In addition, it should be noted that Pepper can be connected according to the ROS communication standard.

TABLE I. EXPERIMENTAL RESULTS.

Result	Count	Ratio[%]
Success	100	**83.3**
Motion planning failure	13	10.8
D-Hand error	4	3.33
Gripping failure	2	1.67
Object detection error	1	0.833

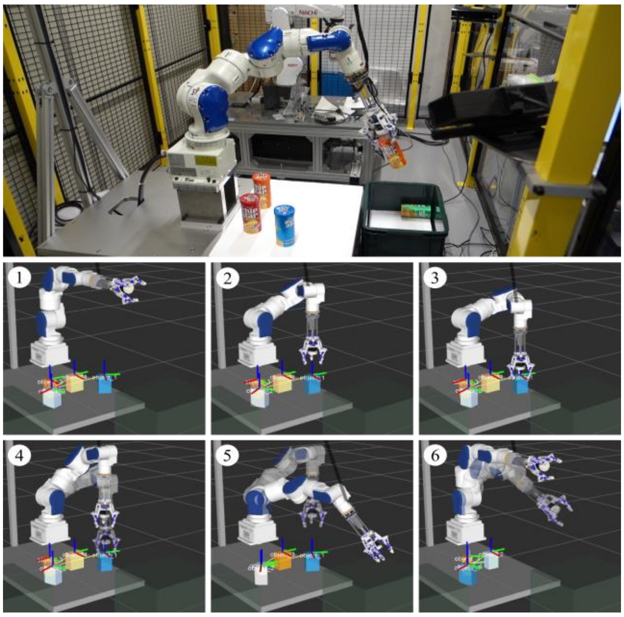


Fig. 7. Motion planning for pick and place task

IV. CONCERNS FOR PRACTICAL USE

We explained the concrete application by taking advantage of the RIS in the previous section. In this section, we discuss the issues and future prospects of practical use of the system.

A. *Hardware configuration of RIS*

We use two Kinects for the RIS in this study. It is necessary to install more sensors in the RIS, because there are insufficient sensors for detecting unexpected events, such as intrusion into the robot operation space, and dropping the grabbed object. However, if the number of installed Kinects increases, the time required for merging the point clouds from each Kinect increases. Therefore, the calculation time (for merging point clouds from each Kinect) becomes significant. Therefore, we need to adjust the point cloud density to reduce the number of points in accordance with the number of installed Kinects.

B. *Calibration of the Kinect*

We must properly calibrate the external parameters to create the merged point cloud accurately. At this stage, it takes a significant amount of time to adjust the position of the camera. Therefore, an application that can instantaneously establish the RIS will be required in the future.

C. *Motion planning algorithm*

Motion planning is calculated using only one frame that is captured before the planning command. The industrial robot cannot operate if the working space situation suddenly changes and it is running according to the previous planned result. In order to enhance the effectiveness of the RIS, the robot must be stopped and the motion trajectory must be re-planned when environmental change occurs [5].

V. CONCLUSION

In this study, we propose a robot intelligent space (RIS) for an industrial robot using two Kinects. Moreover, we confirm the efficiency of the RIS by validating the obstacle detection for motion planning, and pose estimation for grasping objects. In addition, we present an example of a system using the RIS for an industrial robot that does not require any skilled operators; the operation was autonomous. Further, it is possible to apply this method to other robots as it is based on the use of open source software (ROS) and Kinect (which is inexpensive).

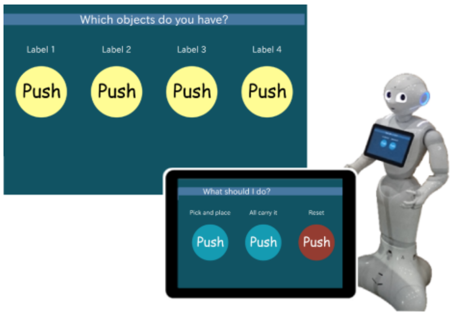


Fig. 8. Manipulation interface using sound guidance and tablet operation of a humanoid robot.

REFERENCES

[1] C. Liu and M. Tomizuka, "Algorithmic safety measures for intelligent industrial co-robots," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, pp. 3095-3102, 2016.

[2] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," in IEEE Robotics and Automation Letters, vol. 2, no. 1, pp. 41-48, Jan. 2017.

[3] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system, " in Proceedings of Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA), 2009.

[4] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, San Francisco, vol. 2, pp. 995-1001, 2000.

[5] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using GPUs," Robotics and Automation (ICRA), 2013 IEEE International Conference on, Karlsruhe, pp. 4090-4097, 2013.