The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Picking up an Indicated Object in a Complex Environment

Kazuyuki Nagata, Takashi Miyasaka, Dragomir N. Nenchev
Natsuki Yamanobe, Kenichi Maruyama, Satoshi Kawabata, and Yoshihiro Kawai

Abstract— **This paper presents a grasping system for picking up an indicated object in a complex real-world environment using a parallel jaw gripper. The proposed grasping scheme comprises the following three main steps: 1) A user indicates a target object and provides the system with a task instruction on how to grasp it, 2) the system acquires geometric information about the target object and constructs a 3D environment model around the target by stereo vision using the information obtained from the task instruction, and 3) the system finds a grasp point based on grasp evaluation using the acquired information. As an example of the scheme, we examined the picking up of a cylindrical object by grasping at the brim. An important and advantageous feature of this scheme is that the user can easily instruct the robot on how to perform the object-picking task through simple clicking operations, and the robot can execute the task without exact models of the target object and the environment being available in advance.**

I. Introduction

One of the most important tasks performed by a service robot is object fetching. The main operation in this task is object picking. However, it is difficult for a robot to automatically pick up an object in a complex real-world environment. This is because there are many different objects that can be grasped and robots as well as humans move such objects in a real-world environment; therefore, it is not realistic to create exact models of the individual objects and the environment in advance. It is also difficult for a robot to identify and recognize an object placed in a complex environment. However, a robot can obtain some information about an object and the environment through a task instruction. Therefore, a system for task instruction is essential for a service robot to execute the object-picking task.

Thus far, many studies have focused on service robots that can pick up an object in a complex environment [1]-[6]. Makihara et al. studied object fetching with a service robot that visually recognizes the target object by engaging in a dialog with the user [1]. Chong et al. used ID tags for identifying the objects to be grasped [4]. Saxena et al. developed a method to find grasp points for novel objects based on a machine learning method [5].

K. Nagata N. Yamanobe, K. Maruyama, S. Kawabata, and Y. Kawai are with the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba 305-8568, Japan {k-nagata, n-yamanobe, k.maruyama, satoshi.kawabata, y.kawai}@aist.go.jp

T. Miyasaka is a graduate student at Tokyo City University, 1-28-1, Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan

D. Nenchev is with Tokyo City University, 1-28-1, Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan nenchev@rls.mse.tcu.ac.jp

We have also constructed a task instruction system that enables a service robot to pick up an indicated object [6]. In this system, the user indicates the target object in a camera image by clicking and then selects the object category and grasp mode. Then, the system receives the 3D position of the chosen point on the target object through the stereo camera, along with the task model for picking up the object with the selected grasp mode. The robot executes the task based on this task model. The system required the user to input the grasp point by clicking on the camera image. However, the user might occasionally miss clicking the grasp point when the target object is far from the camera. Then the system can not determine the 3D position of the clicked point through the stereo camera because no good match can be found. To solve this problem, this paper presents a new grasping system for picking up an indicated object in a complex environment. The proposed grasping scheme comprises the following three main steps:

1) The user indicates a target object and provides the system with a task instruction on how to grasp it.
2) The system acquires geometric information about the target object and constructs a 3D environment model around the target by stereo vision using the information obtained from the task instruction.
3) The system finds a grasp point based on grasp evaluation using the acquired information.

In a study on grasp planning to find a grasp point, Miller et al. introduced the concept of shape primitives and defined a set of grasp starting positions on each primitive for grasp planning [7]. Ekvall et al. studied the planning of an approach vector for object grasping based on human demonstration [8]. Berenson et al. studied grasp planning in a complex environment involving obstacles. They introduced a grasp-scoring function that takes into account the force-closure, the local environment around the object, and the kinematics of the robot [9]. However, grasp planning required exact models of the objects and the environment to be available in advance. Yamazaki et al. studied grasp planning for a mobile manipulator that could acquire an object model using video images [10]. In this case, grasp planning did not require object information to be available in advance. However, it had a different limitation in that the target object had to be the only one in the environment.

Unlike previously proposed methods, our grasping scheme does not require exact models of the object and the environment to be available in advance, and it can realize object picking in a complex real-world environment.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

The remainder of this paper is organized as follows. Section II describes the proposed task instruction scheme. Section III discusses the acquisition of object information and the construction of the 3D environment model around the target. Section IV provides an outline of how to find a grasp point. Section V describes the application of the proposed system to picking up a cylindrical object by grasping it at the brim. Finally, Section VI concludes this paper and describes our future objectives.

II. Task instruction by user

The grasp mode for object picking depends on the object's shape, situation in the environment, and intended purpose. It is difficult for a robot to identify an object and to recognize its situation in a complex environment. In contrast, a human can easily do so and select an appropriate grasp mode. In the proposed task instruction system, the user provides task information to the robot by selecting the object type and an appropriate grasp mode from a pop-up list according to the object's shape, situation, and intended purpose. In this section, we describe the models used for task instruction and the procedure used for providing the task instructions to the robot.

A. Task model

A task model describes the knowledge of a task, i.e., the task parameters that are necessary for executing the task and the template of the action sequence [11]. A task is executed based on the task model according to the robot system. The task models in our task instruction system describe the task parameters, action sequence, and requirement to execute each action for the task of picking up an object.

B. Object model

A real-world environment contains many different objects, and it is not realistic to create models of all such individual objects. However, objects belonging to the same category typically have similar shapes, and these can be handled in the same manner although their designs and sizes might be different. Therefore, we create an object model for an object category. We introduce object primitives having a simple shape for modeling and abstracting the object categories. An object primitive defines the geometric parameters that represent the primitive shape and describes the applicable grasp modes that are linked to the task model for picking up the object primitive with the grasp modes. We define the following seven types of object primitives: a sphere, a tapered cylinder, and a tapered square pillar; a tapered cylinder and a tapered square pillar with a hole; a tapered cylinder and a tapered square pillar with a hole and a bottom. The object model consists of a set of object primitives depending on the object's shape. The object model also has both a typical value and a range of values of the geometric parameters. Figure 1 shows an example of the object model of a cup.

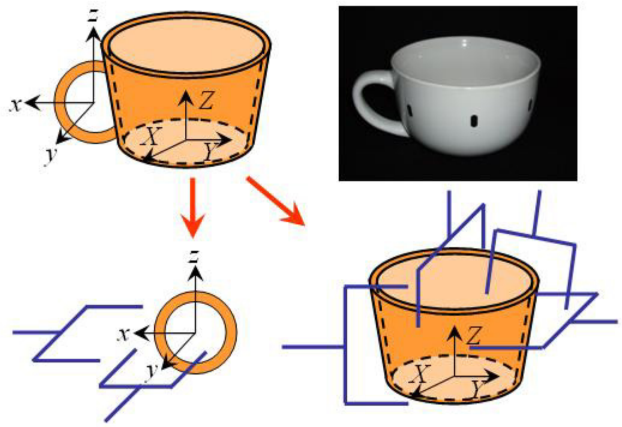


Fig. 1. Object model of a cup. The container of the cup is modeled as a tapered cylinder with a hole and a bottom and the handle of the cup, as a tapered cylinder with a hole. The applicable task models for picking up the primitive are defined in each object primitive.

C. Spatial model

Information about an object that is fixed in an environment, such as a fixture or a piece of furniture, is described as a spatial model, and this spatial model is arranged in a workspace model in the system according to the actual arrangement. When a user customarily places a certain object category in the space available in a fixed object, the object model of the former is registered in the spatial model of the latter. For example, the object model of a dish or a book is registered in the spatial model of a kitchen cabinet or a bookshelf, respectively. In this manner, the workspace model can be customized by registering the object model to the spatial model according to the user's lifestyle [6].

D. System components

The task instruction system consists of a manipulator, a stereo camera, and a monitor. The stereo camera is attached to the wrist of the manipulator. The monitor displays the camera image of the workspace. A range image of the workspace is obtained using the stereo camera.

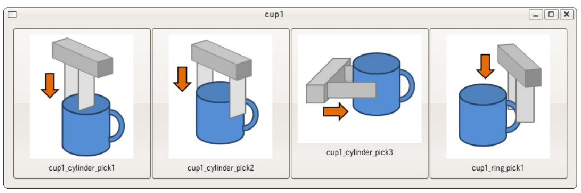
E. Procedure for providing task instruction to robot

The procedure for providing a task instruction to the robot is as follows:

1) The user finds the target object in the camera image and clicks on it (Fig. 2 (a)). The user points at the target object directly, and therefore, the robot does not need to search for it. The system obtains the 3D position of the clicked point on the target object from the range image. The system selects the spatial model in which the clicked point is included, and displays the pop-up list of the object models that are registered in the spatial model (Fig. 2 (b)).
2) The user identifies the target object and selects the object category. The system obtains the object model of the target object and displays the pop-up list of the

(a) Click a target object in camera image

(b) Listing of object models



(c) Listing of grasp modes

Fig. 2. The user indicates the target object and selects the object category and an appropriate grasp mode from the pop-up list for task instruction.

grasp modes that are described in the selected object model (Fig. 2 (c)).

3) The user selects the appropriate grasp mode from the pop-up list according to the object's situation and intended purpose. The system obtains the task model for object picking with the grasp mode.

In this manner, the user provides the task information to the robot through several selection operations on the monitor.

III. Object Information Acquisition and Environment Modeling

From the task instruction, the system receives information about the 3D position of the clicked point on the target object, the spatial model where the target object is placed, the object model of the target, and the task model for picking up the object. The system acquires information about the target object and constructs the 3D environment model around the object with stereo vision using this information.

A. Object information acquisition

Object information such as the value of the geometric parameters and the position and orientation of the object is required for a task. The accuracy of the geometric parameters depends on the grasp mode. For example, Fig. 3 shows four grasp modes for a cylindrical object with a hole. Grasp modes I, II, and III require high accuracy of the radius but not the height. In contrast, grasp mode IV requires high accuracy of the height but not the radius.

The system acquires object information using the model fitting method from the primitive shape of the object and the 3D points obtained by stereo vision.

As an example, we show object information acquisition for a cylindrical object using grasp mode II. In this case, to find the grasp point, information about the brim circle (radius, position, and orientation of the circle) needs to be found with high accuracy whereas that about the height does not need to be found with high accuracy. The system first determines the brim circle and then estimates the object height.

The system detects the brim circle using 3D space stereo vision. The detection procedure is as follows. First, the range image is segmented into some regions by jump edge detection based on the depth discontinuity, and then, the region that includes the clicked point is extracted. This region is considered to be on the surface of the target object. Second, the system identifies the cylinder surface (outer side, inner side, outer bottom, and inner bottom) that includes the extracted region using the mean and Gaussian curvatures, and the rough central axis of the cylindrical object is calculated using normal information of the 3D points in the region. Third, the robot changes its posture to peer into the brim circle using its on-board stereo camera, based on information about the central axis, and captures a 3D image of the brim circle. Fourth, the system reconstructs the 3D boundaries from the stereo image by segment-based stereo vision [12]. Then, the 3D circular arcs that might belong to the brim circle are extracted from the 3D boundaries by comparing the normal vector of the arc boundary and the direction of the central axis. Information about the center of the arc boundary is also used to extract the brim arc. Fifth, the system calculates the brim circle by the least-squares method using the extracted 3D circular arc. The precise central axis is obtained from the brim circle. Figure 4 shows the detection of the brim circle of the cylindrical object.

Next, the system estimates the object height. Figure 5 shows the estimation of the object height. The points in the extracted region that includes the clicked point (Fig. 4 (a)) are projected onto the central axis of the cylindrical object. We define the height of the cylindrical object as the length between the outmost projected point and the center of the brim circle.

B. Environment modeling

The system obtains the spatial model in which the target object is included by the task instruction. However the system does not have information about what is around the object in the spatial model. Therefore, the system constructs the 3D environment model in the spatial model through stereo vision. We use a 3D occupancy grid to construct the 3D environment model [13][14]. The space of the spatial model is discretized into voxels. The voxels can be in one of the following four states: "occupied," "target," "free," and "unknown." The 3D occupancy grid model of the space

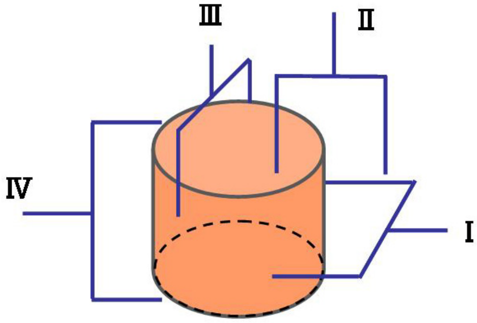


Fig. 3. Four grasp modes for cylindrical object with a hole.

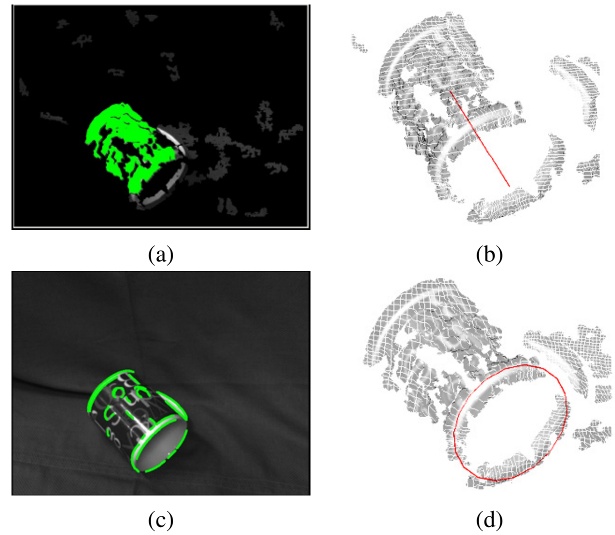


Fig. 4. Detection of brim circle. (a) Extracted region that includes the clicked point. (b) Estimated central axis of the cylindrical object. (c) Circular arcs extracted by segment-based stereo vision. (d) Detected circle of the brim.

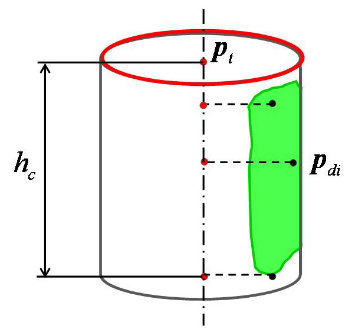


Fig. 5. Estimation of object height.

comprises these voxels. The procedure for constructing the 3D environment model is as follows:

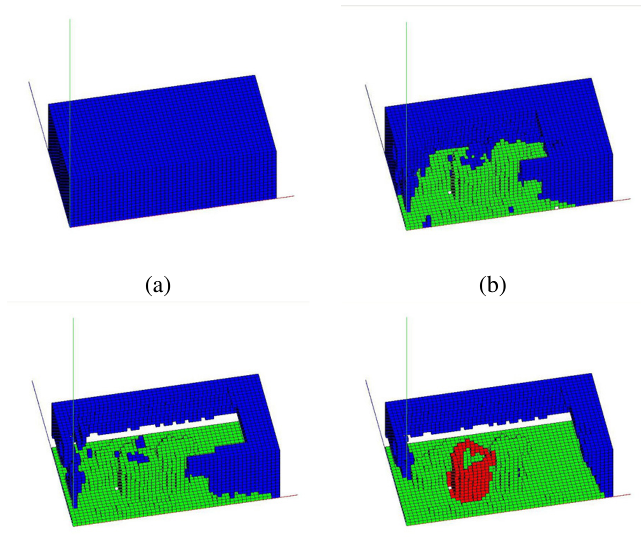
1) Initially, all voxels are considered to be “unknown.”
2) The robot acquires a range image of the workspace using the on-board stereo camera.
3) The voxels that include the 3D point in the range image are considered to be “occupied,” and those on the line of vision between the 3D point and the camera center are considered to be “free.”
4) If the unknown voxels are isolated and surrounded by free voxels, the unknown voxels appear to be floating in the air and are therefore considered to be “free.”
5) Repeat procedures (2) through (4) while changing the posture of the camera.
6) The states of the occupied voxels that include the 3D point on the target object are changed to “target.”

Figure 6 shows the construction of the 3D environment model.

IV. Grasp planning

In this paper, we represent the grasp posture (position and orientation of the hand when grasping an object) as grasp point. The system selects the grasp point using the acquired object information and the 3D environment model. The grasp point should be selected such that all the actions described in the task model can be executed. Our target task is a simple object-picking task that is executed without performing complex manipulation. The object-picking task consists of three actions: approach, grasp, and departure. The details of each action are as given below:

- Approach:
  The approach moment is when the hand approaches the object at a low speed with the fingers open from a point that is some fixed distance away from the grasp point. The approach should satisfy the following conditions:
  - The approach path should be within the movable range of the robot. (Kinematic condition)
  - The robot should not interfere with obstacles along the approach path. (Interference with obstacles)
- Grasp:
  The grasp moment is when the hand closes the fingers in order to grasp the object. The grasp should satisfy the following conditions:
  - The grasp point should be within the movable range of the robot. (Kinematic condition)
  - The robot should not interfere with obstacles at the grasp point. (Interference with obstacles)
  - The fingers should not interfere with the target object before grasping. (Geometric condition)
  - The target object should be between the fingers, and each finger should make contact with the object. (Geometric condition)
  - Force-closure is achieved after grasping. (Force-closure)
- Departure:
  The departure moment comes when the robot moves the grasped object to the global space at a low speed. The departure should satisfy the following conditions:
  - The departure path should be within the movable range of the robot. (Kinematic condition)



(a) (b) (c) (d)

Fig. 6. Incremental construction of 3D environment model. The red, green, and blue voxels represent the target, occupied, and unknown volume, respectively. The volume of the workspace is 600 × 300 × 250 mm. Each voxel is a cube of side 10 mm.

– The robot and the grasped object should not interfere with obstacles along the departure path. (Interference with obstacles)

The grasp point is selected such that all these conditions are satisfied. To select the grasp point, we introduce the following grasp evaluations.

(a) Geometric condition:

The geometric condition imposes constraints on the relative size and position between a hand and a target object that should be satisfied for effective grasping. Examples of such conditions include, the stroke condition of the fingers such that the gripper can grasp the object, the positional condition to avoid interference between a hand and a target object, and the condition for some part of the target object to be grasped between the fingers. The geometric conditions can be described as follows:

$$\boldsymbol{C}_s(\boldsymbol{O}_g, \boldsymbol{H}, G_f) \geq \boldsymbol{0} \tag{1}$$
$$\boldsymbol{C}_p(\boldsymbol{O}_g, \boldsymbol{H}, G_f, \boldsymbol{P}) \geq \boldsymbol{0} \tag{2}$$

where $\boldsymbol{C}_s$ is a vector that combines the size constraints and $\boldsymbol{C}_p$, one that combines the positional conditions. $\boldsymbol{O}_g$, $\boldsymbol{H}$, G_f, and $\boldsymbol{P}$ denote the geometric parameters of the target object, the geometric and mechanical parameters of the hand, grasp mode, and grasp point, respectively. If eq. (1) and (2) are not satisfied, the hand can not grasp the object with the designated grasp mode. Then, the system should inform the user that the robot can not grasp the target object with the designated grasp mode. The parameter ranges of $\boldsymbol{P}$ are calculated from eq. (2), and a set of candidate grasp points is sampled in the parameter space. We introduce the following evaluation for the geometric condition:

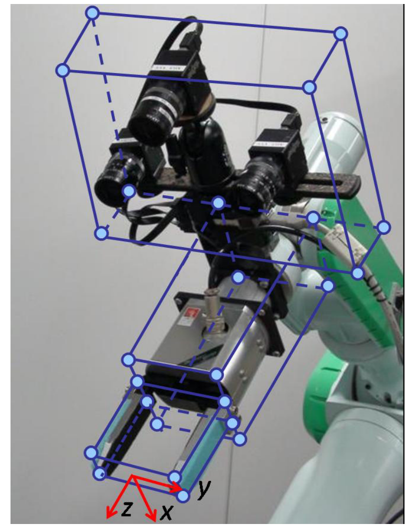


Fig. 7. Model of end-effector with three boxes

$$e_h = \min\ |c_{pi}(\boldsymbol{O}_g, \boldsymbol{H}, G_f, \boldsymbol{P})| \quad (i = 1 \sim n) \tag{3}$$
$$E_1 = e_h / e_{h\_max} \tag{4}$$

where $c_{pi}(\boldsymbol{O}_g, \boldsymbol{H}, G_f, \boldsymbol{P})$ is an element of $\boldsymbol{C}_p$; n, the element count of $\boldsymbol{C}_p$; and e_{h_max}, the observed maximum value of e_h.

(b) Force-closure:

Thus far, many grasp planning methods have used force-closure to evaluate stable grasps [7][9][15]. In this study, as well, we have used the force-closure criterion. We introduce the following evaluation for the force-closure condition.

$$D = \min\ \|\boldsymbol{w}_{Bd}\|, \tag{5}$$
$$E_2 = D / D_{max}. \tag{6}$$

where $\boldsymbol{w}_{Bd}$ is the boundary of the convex hull of contact wrenches; D, the distance between the nearest point on $\boldsymbol{w}_{Bd}$ and the origin [15]; and D_{max}, the observed maximum value of D.

(c) Kinematics:

The system checks whether the manipulator can reach the grasp point. If the check fails, the grasp point is removed from the set of candidate grasp points. Next, the system evaluates the ease of reaching the grasp point by determining the distance between the initial posture and the grasp point. The following evaluations for the position and the orientation are introduced:

$$e_p = 1 - (\Delta p - \Delta p_{min}) / (\Delta p_{max} - \Delta p_{min}), \tag{7}$$
$$e_r = 1 - \Delta\theta / \pi, \tag{8}$$

where Δp and $\Delta\theta$ represent the position and orientation distance, respectively, and Δp_{max} and Δp_{min} represent the observed maximum and minimum values of Δp, respectively.

The following evaluation for the kinematics is introduced:

$$E_3 = \alpha_1 e_p + \alpha_2 e_r, \tag{9}$$

$$\alpha_1 + \alpha_2 = 1, \tag{10}$$

where α_1 and α_2 are weight coefficients.

(d) Interference with obstacles:

The robot should avoid interference with obstacles when executing the task. As shown in Fig. 7, the end effector is modeled by three boxes to check the interference with obstacles. We examine the distance from the three boxes to the occupied and unknown voxels in the 3D environment model. The following evaluation is introduced for checking for interference with obstacles:

$$E_4 = \begin{cases} 1.0 & (l > l_{upper}) \\ (l/l_{upper}) & (l \leq l_{upper}) \end{cases} \tag{11}$$

where l is the minimum signed distance from the three boxes to the occupied and unknown voxels at the grasp point and l_{upper}, the upper limit of the distance at which the robot should consider interference with obstacles while on the move. If E_4 has a negative value, we consider that the robot will interfere with the obstacles.

The procedure for selecting the grasp point is as follows:

1) Sample a set of candidate grasp points in the parameter space calculated by eq. (2).
2) Evaluate the candidate grasp points based on the following evaluation:

$$E = w_1 E_1^2 + w_2 E_2^2 + w_3 E_3^2 + w_4 E_4^2, \tag{12}$$

$$w_1 + w_2 + w_3 + w_4 = 1, \tag{13}$$

   where $w_1 \sim w_4$ are weight coefficients.
3) Select the grasp point among the candidate grasp points that has the highest score in the grasp evaluation.
4) Check the moveable range of the arm and possible interference with obstacles in the approach and the departure paths for the selected grasp point.
5) If the check fails, the grasp point with the next highest score is examined.

V. Experiment

We conducted experiments on picking up an indicated object. As an example, we implemented a task in which the robot picks up a cylindrical object such as a cup or a glass by grasping the brim of the object using a parallel jaw gripper. The target object is modeled as a non-tapered cylinder with a hole and a bottom. Figure 8 shows the geometric parameters of the cylindrical object and the grasp parameters. We set the task frame at the center of the brim circle of the object. The z-axis z_t is set to be along the central axis of the cylinder and is directed into the object. We also set the hand frame at the center between the fingertips. The z-axis of the hand frame is set to be along the longitudinal direction of the finger and is directed outward the hand. The object parameters and the grasp parameters are as follows:

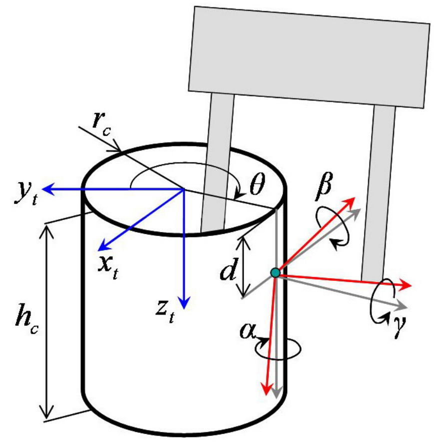


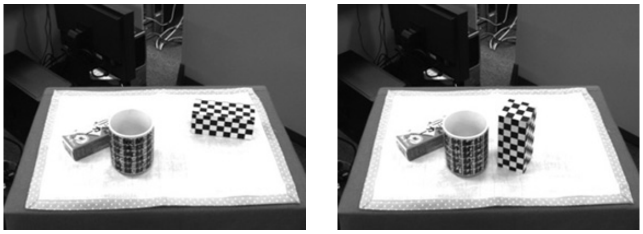
Fig. 8. Grasping at brim of a cylindrical object

- Geometric parameters of the cylindrical object:
  · Position vector of the task frame $\boldsymbol{p}_t$.
  · Rotation matrix of the task frame $\boldsymbol{R}_t$.
  · Radius of the outer circle of the brim r_c.
  · Thickness of the cylindrical object t_c.
  · Height of the cylindrical object h_c.
  · Thickness of the bottom b_c.
- Grasp parameters:
  · Depth of the grasp point from the brim d.
  · Longitude of the grasp point θ.
  · Distance between the central axis of the cylinder and the grasp point r_g.
  · Orientation of the hand at the grasp points α, β, and γ.

The object-picking task is carried out under the following conditions:

1) The thickness t_c and bottom b_c of the vessel have a certain value determined from data for a similar object.
2) The approach path is a straight line and the direction is the same as the z-axis of the task frame. The length of the approach path is fixed.
3) The departure path is a straight line and the direction is upward. The length of the departure path is fixed.
4) The z-axis of the hand frame is the same as the z-axis of the task frame, and therefore, $\beta = 0$ and $\gamma = 0$. In addition, the grip face of the hand is perpendicular to the surface normal of the cylinder at the grasp point, and therefore, the roll angle of the hand α is 0 or π.
5) The manipulator has an initial posture and all tasks are started from this posture.

We use the PA10 robot arm (Mitsubishi Heavy Industries, Ltd.) with a parallel-jaw gripper (Takano Bearing Co., Ltd.) and three IEEE1394 digital cameras with 6-mm lenses (Point Grey Research, Inc.) in the experiment. Figure 9 shows the arrangement of the objects. The target object is a drinking cup having a radius and height of 39 mm and 102 mm, respectively.



(a) (b)

Fig. 9. Arrangement of objects. One small obstacle is placed around the target object and (b) small and large obstacles are placed around the object. The pictures are captured at the initial posture of the manipulator.

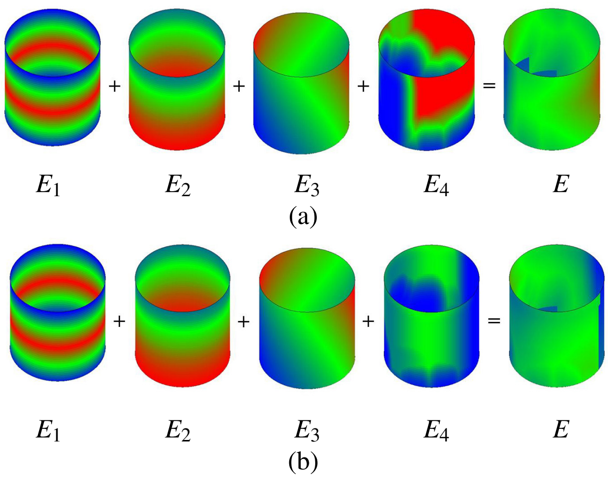
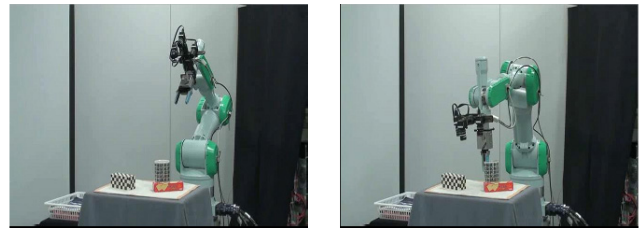


Fig. 10. Grasp score: E_1, E_2, E_3, E_4, and E are scores for the geometric condition, force-closure, kinematics, interference with obstacles, and total, respectively. The red area represents the high score region and the blue area, the low score region.

In the experiment, the user indicated the drinking cup on the camera image, and selected the object category and grasp mode of grasping at the brim. The system acquired the object information and constructed the 3D environment model through stereo vision using the information obtained from the task instruction. The detected radii of the cup are 37.7 mm and 37.4 mm in Fig. 9 (a) and (b), respectively. The estimated heights are 73.2 mm and 76.2 mm, respectively.

Next, the system selected the grasp point using the acquired information and based on the score of the grasp evaluation. The grasp parameters that should be determined are d, θ, and α in accordance with the task specification, where $\alpha = 0$ or $\alpha = \pi$. For the grasp evaluation of the geometric conditions, the system first checks the stroke condition of the fingers, whether the gripper can grasp the object. The system should also satisfy the following geometric conditions:

- The finger should not interfere with the bottom of the object.
- The depth of the grasp point should be shorter than the finger length.
- One of the two fingertips should be inside the cylindrical object.



(a) (b)

Fig. 11. Experimental scene. (a) Initial posture and (b) grasp posture.

(a) (b)

Fig. 12. Experimental scene. (a) Initial posture and (b) grasp posture.

Then, the positional conditions $\boldsymbol{C}_p$ can be written as

$$\begin{pmatrix} d \\ d_{max} - d \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{14}$$

where d_{max} represents the maximum depth of the grasp point calculated by using the geometric parameters of the object and the hand. Then, the parameter ranges of the grasp point are derived by using eq. (14); these are $\theta = [0, 2\pi]$, $d = [0, d_{max}]$, and $\alpha = 0$, π. We sampled and evaluated the grasp point in the grasp parameter space with intervals of $\theta = 10.0^\circ$ and $d = 10.0$ mm, respectively. With regard to the grasp evaluation of the force-closure, we evaluated only the total moment acting on the object. This is because the total finger force remains constant over the depth of the grasp points when the parallel jaw gripper grasps the cylindrical object at the brim with the hand orientation $\gamma = 0$. In the experiment, we set the weight coefficients in both eq. (10) and eq. (13) to the same value. The computation time of the grasp planning was about 25.0 sec on a PC with a 3 GHz dual-core processor. Most of this time was spent on the evaluation of interference with obstacles. Figure 10 (a) and (b) show the grasp score for the cases shown in Fig. 9 (a) and (b), respectively. The viewpoint of each picture is the same as that in Fig. 9. Figures 11 and 12 show the experimental scene when the objects are placed as shown in Fig. 9 (a) and (b), respectively. The experiments indicated that the robot can select the grasp point while avoiding interference with obstacles.

VI. Conclusion

In this paper, we presented a grasping system for picking up an indicated object using a parallel jaw gripper in a complex environment. The proposed grasping scheme comprises the following three main steps. First, the user indicates a target object and provides the system with a task instruction on how to grasp it. Then, the system acquires information about the 3D position of the clicked point on the target object, the spatial model where the object is placed, the object model of the target, and the task model for picking up the object. Second, the system acquires geometric information about the target object and constructs a 3D environment model around the target by stereo vision using the information obtained from the task instruction. Third, the system finds a grasp point based on grasp evaluation using the acquired information. We evaluated the grasp quality as a geometric condition, force-closure, kinematics of the manipulator, and interference with obstacles. As an example of the scheme, we examined the picking up of a cylindrical object by grasping at the brim. An important and advantageous feature of this scheme is that the user can easily instruct the robot on how to perform the object-picking task through simple clicking operations, and the robot can execute the task without exact models of the target object and the environment being available in advance.

In future work, we intend to apply our grasping scheme to a greater variety of objects and more complicated real-world environments. We plan to develop a view planning scheme for acquiring object information in a cluttered environment, and we also plan to use a tactile sensor to supplement visual sensing.

REFERENCES

[1] Y. Makihara, M. Takizawa, K. Ninokata, Y. Shirai, J. Miura and N. Shimada, "A Service Robot Acting by occasional Dialog - Object Recognition Using Dialog with User and Sensor-Based Manipulation -," *Journal of Robotics and Mechatronics*, Vol. 14, No. 2, pp. 124–132, 2002.

[2] F. Saito and T. Suehiro, "Toward Telemanipulation via 2-D Interface - Concept and First Result of "Titi" -," *Proc. of IECON 02*, Vol.3, pp. 2243-2248, 2002.

[3] J. Neubert, P. Ravindran and N. J. Ferrier, "An Assistive Robot Interface for an Interactive Visually Guided Grasping System," *Proc. of the ICORR 2003*, pp. 290–293, 2003.

[4] N. Y. Chong, H. Hongu, K. Ohba, S. Hirai and K. Tanie, "A Distributed Knowledge Network for Real World Robot Applications," *Proc. of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 187–192, 2004.

[5] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic Grasping of Novel Objects using Vision," *International Journal of Robotics Research*, Vol. 27, No. 2, pp. 157–173, 2008.

[6] K. Nagata, Y. Wakita and E. Ono, "Task Instruction by Putting Task Information in Work Space," *Proc. of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 305-310, 2007.

[7] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," *2003 IEEE Int. Conf. on Robotics and Automation*, pp. 1824–1829, 2003.

[8] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," *2007 IEEE Int. Conf. on Robotics and Automation*, pp. 4715–4720, 2007.

[9] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, J. Kuffner, "Grasp Planning in Complex Scenes," *Proc. of Int. Conf. on Humanoid Robots*, pp. 42–48, 2008.

[10] K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, "A Grasp Planning for Picking up an Unknown Object for a Mobile Manipulator," *Proc. of 2006 IEEE Int. Conf. on Robotics and Automation*, pp. 2143–2149, 2006.

[11] K. Ikeuchi and T. Suehiro, "Towards an assembly plan from observation: Part 1: Assembly task recognition using face-contact relations (polyhedral objects)," *Proc. of 1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2171–2177, 1992.

[12] Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita, "3D Object Recognition in Cluttered Environments by Segment-Based Stereo Vision," *Int. J. of Computer Vision*, vol. 46, No. 1, pp. 5–23, 2002.

[13] A. Elfes: "Sonar-Based Real-World Mapping and Navigation," *IEEE J. of Robotics and Automation*, Vol. 3, No. 3, pp. 249–265, 1987.

[14] A. Nakhaei, and F. Lamiraux, "Motion Planning for Humanoid Robots in Environments Modeled by Vision," *Proc. of the 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 197–204 2008.

[15] C. Ferrari, and J. Canny, "Planning Optimal Grasps," *Proc. of 1992 IEEE Int. Conf. on Robotics and Automation*, pp. 2290–2295, 1992.