Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Handling of Objects with Marks by a Robot

Rie KATSUKI[1], Jun OTA[1], Yusuke TAMURA[1], Takahisa MIZUTA[1], Tomomi KITO[1], Tamio ARAI[1], Tsuyoshi UEYAMA[2], and Tsuyoshi NISHIYAMA[2]

[1] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
[2] DENSO WAVE Inc., 1-1, Showa-cho, Kariya, Aichi 448-8661, Japan
rie@prince.pe.u-tokyo.ac.jp

Abstract

This paper presents a robot system for handling various objects in home or office environments (see video). A fixed manipulator utilizes marks on objects for handling and motion planning. A mark consists of two parts. One is the outer part, which indicates its pose (In this paper, "pose" denotes 3D position and orientation.), and the other is a QR code, which is a kind of 2D barcode. A QR code is stored information of an object (e.g. its name). A user attaches several marks on each object. The manipulator accesses the information using a camera in its hand. The robot can decide its complex handling motion based on the information in QR codes, pose of the objects, which are estimated from pose of marks, and signals from proximity sensors. Experiments are conducted to verify the whole system.

Key Words: Manipulation, Motion planning, Artificial mark, Sensor, Measurement

1 Introduction

Robots are becoming increasingly important both in home and office environments. Handling of objects is one of the most important tasks that the robot executes. To accomplish the task, we have to treat two phase: (a) recognition of various objects in actual time, and (b) motion planning based on the poses of the objects (In this paper, "pose" denotes 3D position and orientation.).

In phase of (a), many recognition methods have been proposed from both model-based method and appearance-based method. For example for model-based method, Boshra *et al.* have recognized partially occluded polyhedrons through multi-stage matching process [1]. Sumi *et al.* have proposed a method that recognizes free-form objects using the 3D pose of an object outline for matching [2]. For example for appearance-based method, Chen *et al.* have recognized partially occluded objects by using edges inside outline of a object for aspect representation, and heuristic function [3]. However, they have not treat a case when distinctive parts of objects are occluded. Moreover, in home or office, there are many objects for handling by a robot, many models are needed to object matching. Computational cost is naturally high to match an object with the many models. Therefore, (a) has not been solved yet.

In phase (b), There are bin-picking systems that consider collision between a robot and other objects [4]. However, the systems cannot be applied in home or office environments because the system can handle a few variations of form, and objects that could cause mistakes in recognition must be removed in advance. Rössler *et al.* have planned grasp points for unfamiliar objects by learning [5]. Terasaki *et al.* have proposed intelligent manipulation methods [6]. It used custom-designed robot hand and state-space approach to make complex manipulation possible. However, [5] solved the problem under an assumption that limits observing poses. [6] needed an accurate map about objects and environments. However, objects recognition is not reliable to build an accurate map in general. Petersson *et al.* have performed handling task by fusing image processing and pre-computed paths [7]. However, the grasp poses were decided in advance, and a manipulator could not change its grasp pose in consideration of collision with other objects. Therefore, (b) has not been solved yet.

In this paper, we propose a robot system for objects handling. The robot recognizes various objects, and then, it adopts suitable grasp poses and sequences of conveyance, which depend on the various arrangements of objects. To accomplish this purpose, we have adopted marks [8]. These marks contain information that is requited by the robots for successful handling (e.g. the object's name and storage area and the grasp pose candidates for the robot hand). A robot can use this information to handle an object successfully. In Sections 2, System overview is explained. Method to calculate an object pose from some marks poses is described in Section 3. Then, motion planning using the marks is described in Section 4. Finally, we verify the validity of the proposed method through a handling experiment in Section 5.

0-7803-7860-1/03/$17.00 © 2003 IEEE

2 System Overview

The system consists of a fixed manipulator, which is called a robot in this paper. This robot handles objects that are put on a table. It is equipped with a parallel-jaw gripper. There are proximity sensors on this hand. The robot can change its pose of view by a camera on the hand.

Fig. 1 shows an overview of the proposed robot system. Firstly, a user makes the marks and attaches them to objects. The mark consists of two parts. One is the outer part, which indicates its pose, and the other is a QR code, which is a kind of 2D barcode. The mark is introduced for the following reasons. (a) It guarantees accuracy of positioning when a robot handles objects (the translation is 15mm, and the rotation is 15deg)[8]. (b) It can be attached on a curved surface. (c) A user can create a mark easily using a color printer. (d) Even if the mark rotates around its normal vector, the QR code on the mark still presents its information to the robot. (e) If a robot has a zoom CCD camera, it has a wide area for marks measurements.

A robot measures pose of marks by color extraction from one camera image, then it decodes QR codes. As mentioned in Section 1, the information stored in a QR code is unique to a particular object. This information is made by a user. A robot can handle various objects using this information and the pose of marks (see Fig. 1-3). The marks have the following advantage. A robot can recognize any objects by simple image processing equipment and algorithm. Using the explicit information about objects, which is stored in marks, the robot can easily select an appropriate grasp pose.

The success rate of handling greatly depends on detection rate and measurement accuracy of marks; consequently, a robot should detect a mark and measure its pose with high accuracy at all times. In our method, these requirements are complied with by attaching some marks to one object. In this case, a robot has to calculate one object pose using poses of the marks that are attached to the object and have measurement errors. We therefore calculate the object pose by weighted least-squares method. The weighting value is decided by an estimated standard deviation from measurement error of a mark. Measured marks poses with high error are eliminated through the weighting factors; therefore, the robot can calculate an object pose with low error (see Section 3).

When a robot plans its grasp pose, it has to check collision between its hand and other objects. The collision check is executed based on the candidate of grasp poses in QR codes. A robot can decide a suitable grasp pose from the candidates in consideration of other object arrangements (see 4.1-4.3). If some of marks are not detected by occlusion, a robot re-determines its grasp pose using a signal from proximity sensors (see 4.4). This process increases the success rate of handling.

3 Method to Calculate an Object Pose

3.1 Procedure for the Estimation of Reliability for Marks Poses

There is a correlation between reliability of measured marks poses and deviations of error on the marks poses. The deviations depend on the marks shape in a camera image, i.e., marks poses in a camera coordinate system; therefore, we adopt marks poses to estimate the reliability. For future discussion, we define two assumptions. (1) A measured pose follows a normal distribution. (2) Deviation is explained through standard deviation.

To estimate the reliability, we prepare StdDev maps. These are 3D graphs that are plotted standard deviation of measurement errors (describe later). Ideal StdDev map should present six standard deviations (σ_x, σ_y, σ_z, σ_{rx}, σ_{ry}, σ_{rz}) using six parameter of a mark pose (x, y, z, r_x, r_y, r_z). However, we simplify the maps to reduce data amount. After this, we describe a method to simplify the maps with minimize errors of standard deviations.

A coordinate system can be seen in Fig. 2, method to

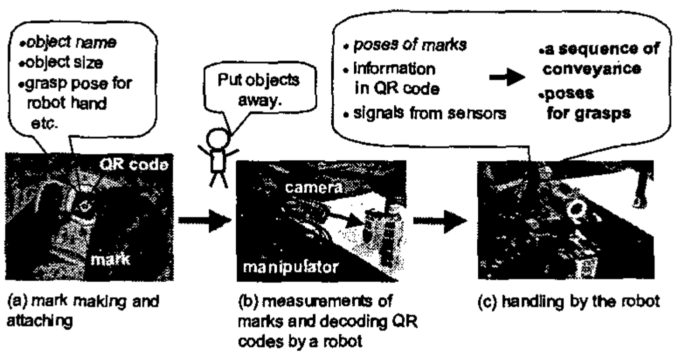


Fig. 1 Concept of the handling using the marks

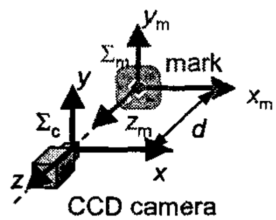


Fig. 2 Coordinate system

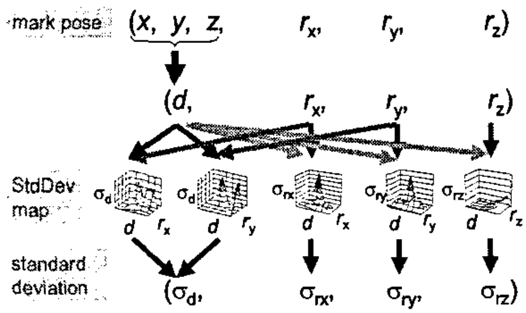


Fig. 3 Estimation of reliability of a mark pose

estimate reliability of marks poses can be seen in Fig. 3. First, a robot measures a mark, and obtains the mark pose (x, y, z, r_x, r_y, r_z) in a camera coordinate system. The pose is expressed in Roll-Pitch-Yaw angles. At actual mark measurement, the robot zooms in since it enable to measurement from a distance. In this case, the mark has to keep remaining in the camera image; thus, the robot moves the camera to a place where the mark is placed in the camera center. We consequently convert translation (x, y, z) into d - Euclidean norm between the origin on a mark coordinate and optical center of the camera -; calculation for the reliability will be easily.

The reliability is presented as a standard deviation of an error of a mark pose. The standard deviations are stored in StdDev maps. These maps in advance store the standard deviations of the errors that are measured various observation poses, i.e., the maps are 3D graphs that consist of d: a distance axis, θ: a rotation axis, and σ: standard deviation of the pose error that is measured at (d, θ). θ stands for a component of rotation: r_x, r_y, or r_z, which is a orientation from a camera to a mark. σ stands for standard deviation of the distance: σ_d, or rotation: σ_{rx}, σ_{ry} or σ_{rz}.

Ideal StdDev maps should present each σ using (d, r_x, r_y, r_z); however, many data are needed to create the maps. On the other hand, accuracy of a mark pose for handling is not strict (see Section 2). Therefore, we define two assumptions and estimate each σ from only (d, θ). (1) Each components of (d, r_x, r_y, r_z) are not influenced by other components. (2) An observation pose is a discrete value. At measurement to create the maps, the other components are fixed on 0deg. The range of the axis d is 150-400 mm, and its pitch is 50 mm. The range of the θ axis is 0-90 deg as an absolute value, and their pitches are 20 deg. The ranges of d and θ axis are decided by measurable range of a mark. An intermediate value of σ is estimated by linear interpolating.

A robot has a StdDev maps in advance. At handling, the robot can obtain standard deviation $(\sigma_d, \sigma_{rx}, \sigma_{ry}, \sigma_{rz})$ by that it assigns a mark pose(d, r_x, r_y, r_z) to the maps. As shown in Fig. 3, each map is needed to estimate σ_{rx}, σ_{ry}, or σ_{rz}; in contrast, σ_d needs two maps. The two maps are used based on a nature that a steep observation angle causes a high error of mark pose. For the estimation of σ_d, a larger value of r_x and r_y is adopted for the rotation axis. r_z is not considered since a change of r_z does not change the shape of a mark in an image.

3.2 Procedure for the Calculation of an Object Pose

A robot measures marks and acquires N mark poses $\boldsymbol{p}_i = [p_{xi}, p_{yi}, p_{zi}, p_{rxi}, p_{ryi}, p_{rzi}$ (i=1, 2, ..., N) and standard deviation of measuring error $\boldsymbol{\sigma}_i = [\sigma_{di}, \sigma_{rxi}, \sigma_{ryi}, \sigma_{rzi}]$ (i = 1, 2, ..., N), which correspond to the mark poses, from the StdDev maps. With this mark, a standard deviation becomes small if the observation distance or orientation is small; naturally, the weighting factor becomes large. If there is an error that makes the measured value small, this weighting factor is not appropriate. Consequently, we change the weighting factor into suitable value as following: first, a mark is measured several times at a same observation pose, second, the measurement poses are merged using a simple average.

When searching of marks, a robot moves and measures marks at some observation poses; therefore, pose of the camera coordinate systems in a robot coordinate system is different each other. Moreover, to handle objects, not mark poses but object poses are needed. Therefore, these camera coordinate systems should be translated to one coordinate system. $\boldsymbol{p}_i$ and $\boldsymbol{\sigma}_i$ are translated to poses $\boldsymbol{q}_i = [q_{xi}, q_{yi}, q_{zi}, q_{rxi}, q_{ryi}, q_{rzi}]$ (i = 1, 2, ..., N) and standard deviation $\boldsymbol{\delta}_i = [\delta_{di}, \delta_{rxi}, \delta_{ryi}, \delta_{rzi}]$ (i = 1, 2, ..., N) in the robot coordinate system. The translation of $\boldsymbol{\delta}_i$ is based on the error propagation law [9]. δ_i is defined as weighting factor; an object pose with minimum error $\hat{\boldsymbol{q}}$ is calculated using the weighted least-squares method. This is done for each component of the pose. A calculation equation about r_x can be seen in Eq. (1). As well as Eq. (1), other components of the pose can be calculated.

$$\hat{q}_{r_x} = \sum_{i=1}^{N}\left(q_{r_x i}/\delta_{r_x i}^{\ 2}\right) \Big/ \sum_{i=1}^{N}\left(1/\delta_{r_x i}^{\ 2}\right) \quad (1)$$

4 Procedure for Decision of a Grasp Pose

4.1 Expressive Form of Objects

A unique coordinate system for each object is defined as homogeneous transformation matrix. The matrix expresses a pose from an origin of a mark to an origin of an object. It is stored in the mark (see "origin" in Fig. 6). The grasp pose candidates and an object shape that is approximated as a rectangular parallelepiped are also stored in the mark (see "handle_pose" and "shape" in Fig. 6). The former is expressed as a hand pose - a homogeneous transformation matrix, which expresses a pose from an origin of an object to an origin of a hand-. The latter is expressed as (x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}), which is a size in the object coordinate system. This expressive form is reasonable, since the object shape is only used when deciding the conveyance order and the grasp poses, and since an easy description is desirable for non-professional users. In addition, a shape and a coordinate system of a table on which objects are put are also defined through a mark.

4.2 Deciding the Conveyance Order

If a robot approaches an object in consideration of only Euclidean norm from the robot to the object, the robot may collide to other objects located in front of the target. Consequently, we propose following process. First, the robot defines an overlapping line with the axis that first intersects with the robot in the table coordinate system; then, the line is moved in parallel until it intersects with an object; last, the robot approaches the intersected object. In addition, we define a threshold to judge whether objects overlap or not. The threshold is decided from the size of the hand. Concretely, we judge that the objects have overlapped when a distance from a line to a next line is within the threshold. The first line is intersecting an object; the next line is intersecting another object. At handling, if the robot judges as a overlapping, it approaches from an object with high arrangement height.

4.3 Deciding the Grasp Pose

The robot checks the collision between the hand and the objects or the table. In this calculation, the shape of the hand is approximated as a rectangular parallelepiped. The robot has a model of the hand in advance since the shape of hand has not been changed. The robot checks the intersection of a line of the rectangle parallelepiped and the face of another rectangle parallelepiped [10]. If all grasp pose candidates will collide to the other objects, the robot changes conveyance order of objects, which swaps the target for a next object.

4.4 Changing the Grasp Pose using Proximity Sensors

If a robot cannot detect marks on objects, it decides a grasp pose without consideration for the objects. To detect unknown objects, as can be seen in Fig. 4, proximity sensors on the tip of the hand are applied to check signals from sensors while approaching a target. If the robot detects unknown objects from the signal, it moves to the next grasp pose candidate. Here, to detect unknown objects certainly, we define a rule that the tips of the fingers should approach an object firstly.

5 Experiment

We conducted an experiment of object handling by a robot to verify the proposed method.

5.1 Experimental setup

We applied a fixed 6-d.o.f.manipulator DENSO AM-60A0D and a parallel-jaw gripper (opening width, 0-100mm). The manipulator moves 70mm/s, and the hand moves 10mm/s for safety. As can be seen in Fig. 4, the proximity sensors are fixed on the hand. We also applied a QR code reader KEYENCE TL-600, a CCD camera SONY FCB-1X10, an image processing board Hitachi IP5005, and a computer (Pentium III, 1GHz). A white LED light was located on the tip of the camera so that the lighting conditions would be stable.

Fig. 5 shows the experiment setup. There are three targets on the table: the 500ml plastic bottle, the 250ml juice pack, and the flashlight. The robot conveys them to container box 1 or 2. As mentioned in Section 2, the QR code in a mark is written information. For instance, Fig. 6 shows part of the information for the plastic bottle. In this case, six kinds of grasp pose candidates are stored. A user decides the candidates. Just for reference, some of grasp pose candidates for the plastic bottle and the juice are shown in Fig. 7.

As can be seen in Fig. 8, this object arrangement makes handling difficult. First, the robot has to recognize the overlapped objects individually. The

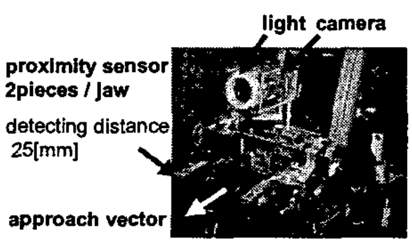


Fig. 4 Structure of the hand and sensors

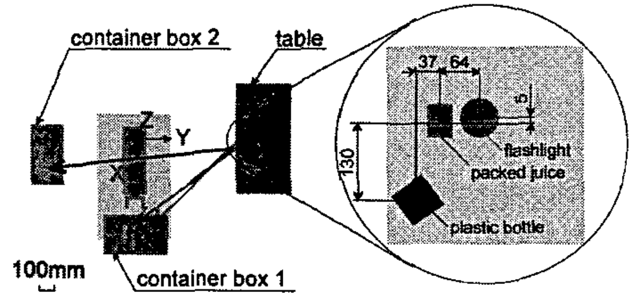
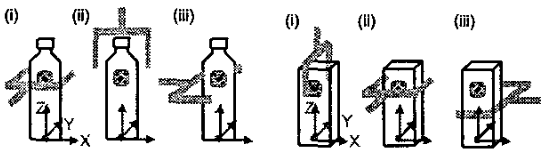


Fig. 5 Experimental setup

```
shape:30,-30;30,-30;210,0
name:*******(500ml)_1
handlle_pose:-1,0,0;0,0,1;0,1,0;0,0,130
handle_pose:-1,0,0;0,1,0;0,0,-1;0,0,130
handle_pose:0,-1,0;-1,0,0;0,0,-1;0,0,130
handle_pose:1,0,0;0,0,1;0,-1,0;0,0,80
handle_pose:0,-1,0;0,0,-1;-1,0,0;0,0,80
handle_pose:0,1,0;0,0,1;1,0,0;0,0,80
origin:1,0,0;0,0,-1;0,1,0;0,-120,-30
```

Fig. 6 QR code data (plastic bottle)



(a) plastic bottle (b) juice pack

Fig. 7 Grasp pose candidates

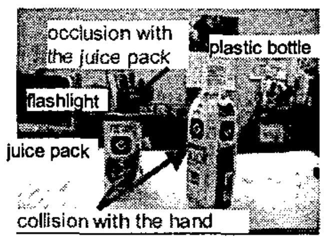
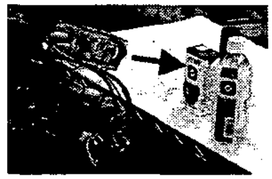


Fig. 8 Object arrangement

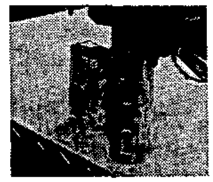
juice pack hides the flashlight from the robot's view; therefore, it is quite hard for the robot to differentiate between the juice pack and the flashlight. Moreover, the robot has to decide a suitable conveyance order and its grasp pose to avoid collision with other objects. For instance, when approaching the juice pack, the robot should choose a grasp pose that does not collide with either the plastic bottle or the flashlight. Furthermore, the mark on the flashlight is occluded; then, the robot has to plan its grasp pose without knowing the pose of flashlight. The robot must plan its suitable grasp pose through the above objects arrangement.

5.2 Experimental results

Fig. 9 shows experimental images. (1) The robot searched and measured the marks, and then it estimated the objects poses (Fig. 9-a). The flashlight had not been detected yet due to a mark occlusion caused by the juice pack. (2) The robot decided a conveyance order. The first target was the plastic bottle. The second was the juice pack. (3) The robot decided a grasp pose for the plastic bottle. It checked collision with grasp pose candidate (i), which is shown in Fig. 7-a, and other object; however, as can be seen in Fig. 10, it judged that the candidate must cause collision with its hand and the

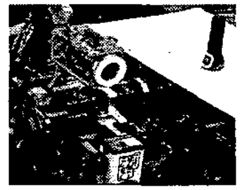


(a) searching the marks



(b) grasp of a plastic bottle

(c) collision check by sensors



(d) handling a juice

Fig. 9 Experimental images

juice pack. Thus, it checked a next candidate (ii), and then it grasped a plastic bottle, as shown in (Fig. 9-b), and conveyed the bottle. (4) The robot decided a grasp pose for the juice pack. It moved to the grasp pose (i) shown in Fig. 7-b; it might have collided with the flashlight if it continued approaching, since it had not recognized the flashlight. However, the robot could detect the flashlight with the proximity sensors (Fig. 9-c); then, it changed its pose to the grasp pose (ii) and handled the juice pack (Fig. 9-d). (5) The robot searched on the table again to check whether there were remained objects or not. Then, it detected and handled the flashlight.

With 15 trials, the robot could handle all objects 13times. Causes of two failures were collision between the hand and the flashlight. The following is details of the failures. First, the pose error of the juice caused the pose error of grasp pose. Second, unfortunately, the robot approached a place where flashlight is in a blind spot of sensors. Then, the sensors did not work. Therefore, the robot collided with the flashlight. The worst error among the translation components in these trials was the y component (at the robot coordinate system shown in Fig. 5). The value was 20.9mm. The worst error among rotation components was 5.2deg at r_z. The most likely reason for these results was observation of all marks from steep poses. These poses caused high errors of the marks poses. Then, the errors were not canceled with the weighted least-squares method. However, this problem can be solved easily by addition of sensors.

Just for reference, absolute values of average measurement errors and their standard deviations of y and r_z are shown in Fig. 11. An error of y at flashlight is high, but then, its standard deviation is low. Therefore, a reason for the error is that a mark seems to

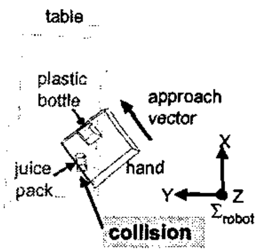


Fig. 10 A result of collision check

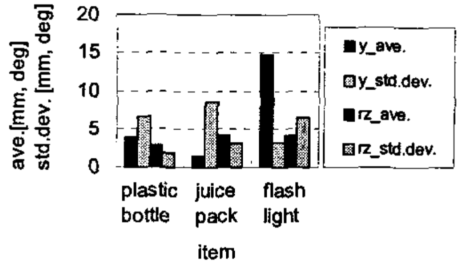


Fig. 11 Calculation results of objects

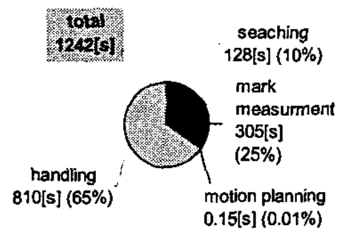


Fig. 12 A breakdown of the work time

be smaller than its real size by curvature of flashlight. Although, the error is smaller than the required accuracy for handling which is described in Section 2, so that the robot could handle the flashlight.

Average work time was 20 minutes and 42 seconds in the success case. A breakdown of the work time is shown in Fig. 12. As mentioned in the Section 3, in mark searching and measurement, the robot had been moving the camera to a place where the mark was placed in the camera center. Mark measurement time that the moving time was cut out is 0.73s×37=27.1s(Ave.), barcode measurement time is 0.05s×27=1.35s(Ave.). If we assume the speed as 500mm/s, which is an appropriate speed in a home or office environment, the estimated work time is about 200s. Both the success rate and estimated work time are feasible; therefore, the proposed method has usability.

6 Conclusion

In this study, we have present a robot system for handling objects in home or office environments. Even if objects are occluded, the system should recognize objects and decide its motion from the objects arrangement. Our approaches for these problems are as follows.

Approach 1: Marks are attached on objects. They are used to provide information such as marks pose and grasp pose candidates; the robot can handle objects. Approach 2: To reduce a case where no mark is observed, more than two marks are attached on one object. If the robot observes more than two marks, the weighted least-squares method is introduced. To decide weighting factor, we propose and prepare the StdDev map. Approach 3: The robot plans grasp poses based on the marks, which are stored a shape of an object and grasp pose candidates for an object. When the robot moves, proximity sensors are utilized to detect an unknown object; the robot can change its motion if it detects the object.

In the experiment, a robot could handle three objects that were occluded or were close each other. The robot chosen a suitable motions using the object information in the marks and signals from the proximity sensors, succeeding 13times out of 15 times.

As future work, we will develop equipment that can automatically create object information for a QR code in a mark.

References

[1] Michael Boshra and M.A. Ismail: "Recognition of occluded polyhedra from range images," Pettern Recognition 33pp.1351-1367, 2000.

[2] Yasushi Sumi and Fumiaki Tomita: "3D Object Recognition Using Segment-Based Stereo Vision," 3rd Asian Conference on Computer Vision, ACCV98, pp.249-256, 1998.

[3] Jin-Long Chen and George C. Stockman: "Determining Pose of.3D Objects With Curved Surfaces," IEEE Trans. Pattern Anal. & Machine Intel. Vol.18, No.1, pp.52-57, 1996.

[4] Krisnawan Rahardja and Akio Kosaka: "Vision Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cues," Proc. IROS 96, pp.1448-1457, 1996.

[5] Bernd Rössler, et al.: "Visual Guided Grasping of Aggregates using Self-Valuing Learning," Proc. IEEE Int. Conf. Robotics & Autom., pp.3912-3917, 2002.

[6] Hajime Terasaki and Tsutomu Hasegawa, "Motion Planning of Intelligent Manipulation by a Parallel Two-Fingered Gripper Equipped with a Simple Rotating Mechanism," IEEE Trans. Robotics & Autom., Vol.14, No.2, pp.207-219, 1998.

[7] L. Petersson, et al.: "Systems Integration for Real-World Manipulation Tasks," Proc. IEEE Int. Conf. Robotics & Autom., pp.2500-2505, 2002.

[8] Rie Katsuki, et al.: "Design of an Artificial Mark to Determine 3D Pose By Monocular Vision", Proc. IEEE Int. Conf. Robotics & Autom., to appear, 2003.

[9] Richard P. Paul: "ROBOT MANIPULATORS," The MIT Press, 1981.

[10] John W. Boyse: "Interference Detection Among Solids and Surfaces," Communications of the ACM, Vol.22, No.1, pp.3-9, 1979.