Pick and Place Planning for Dual-Arm Manipulators

Kensuke Harada, Torea Foissotte, Tokuo Tsuji, Kazuyuki Nagata,
Natsuki Yamanobe, Akira Nakamura, and Yoshihiro Kawai

***Abstract*— This paper proposes a method for planning the pick-and-place motion of an object by dual-arm manipulators. Our planner is composed of the offline and the online phases. The offline phase generates a set of regions on the object and the environment surfaces and calculates several parameters needed in the online phase. In the online phase, the planner selects a grasping pose of the robot and a putting posture of the object by searching for the regions calculated in the offline phase. By using the proposed method, we can also plan the trajectory of the robot, and the regrasping strategy of the dual-arm. Here, the putting posture of the object can be planned by considering stability of the object placed on the environment. The effectiveness of the proposed method is confirmed by simulation and experimental results by using the dual-arm robot NX-HIRO.**

I. INTRODUCTION

The action of picking an object, moving and placing it at a desired location is one of the most typical tasks a robot is required to achieve. However, currently in automated production processes, pick-and-place motions are usually computed offline and thus cannot adapt efficiently to small unpredicted modifications of the conditions. This research tackles the problem of generating the pick-and-place motion of a known object in real-time. When planning the pick-and-place motion, we have to successively plan (1) the hand pose to pick up the object, (2) the hand pose to place the object on the environment at the desired location, (3) the stable pose of the object in the environment, (4) the trajectory of the robot arms to achieve the pick-and-place task, and (5) the regrasping strategy, if it is needed. However, there have been no research on the pick-and-place planner by solving the above planning problems at the same time. Especially, if the shape of the grasped object is complex and if the object is placed in a complex environment, it becomes difficult to find the posture of the hand stably grasping the object and the stable posture of the object placed on the environment.

The authors have proposed a grasp planning approach for parallel grippers[15] where the posture of the hand stably grasping the object is computed by taking the elasticity of the finger surface into consideration. This method is effective since we can estimate the contact area between the finger and the object and can plan a grasping posture with large contact area. By extending this result, this research proposes a pick-and-place planner which consider all the necessary steps required to take, move and release an object using a dual-arm manipulator(Fig. 1).

Fig. 1. Pick-and-place task by a dual-arm manipulator

Our proposed planner is composed of the online and the offline phases. In the offline phase, we obtain a set of regions on the object and the environment surfaces by clustering the polygon models. To find stable postures of the object placed on the environment, the 2D bounding box and the boundaries of the regions are obtained. In the online phase, by considering the relative position of the bounding box between the object and the environment, we calculate the stable posture of the object. Here, we consider the face-face and the edge-face contact between the object and the environment. In the online phase, we also calculate the grasping posuture of the robot. The grasping posture is calculated by considering the pose of the object when picking it up and when placing it on the environment. To plan the grasping posture, we also consider the trajectory of the pick-and-place motion. We further search for the regrasping strategy. We prepare multiple strategies give priorities to them. We consider switching the strategy if the motion cannot be planned with the strategy of the higher priority.

This paper is organized as follows: after introducing the related works in Section 2, Section 3 details the offline and online processes of our pick-and-place planner. Finally, Section 4 demonstrates the efficiency of our method through several numerical examples and experimental results.

II. RELATED WORKS

There have been many researches on grasp planning so far. Borst et al.[1], [2] and Miller et al.[3] proposed grasp planning methods for multi-fingered hands. Diankov[10]

K.Harada, K.Nagata, N.Yamanobe, A.Nakamura, and Y.Kawai are with Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan `kensuke.harada@aist.go.jp`

T.Tsuji is with the Faclty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan `tsuji@ait.kyushu-u.ac.jp`

proposed the grasp planning method considering the quality of the grasping posture. The authors have also researched the grasp planning for multi-fingered hands [12], [13] by assuming the rectangular-convex model for both grasping region of the hand and the grasped object and have proposed the grasp planning for parallel grippers taking the elasticity of the finger into consideration[15]. Here, the number of research on grasp planning considering the putting posture of the hand is limited. Berenson et al.[9] planned the grasping posture of a multi-fingered hand taking the putting posture into consideration. Here, the putting posture of the object is assumed to be fixed.

As for the research on the bin-picking, Ikeuchi[8], Ghita et al.[7] and Dupis et al.[5] proposed methods for picking the top most object. However, the above researches plan the posture of the hands only by considering the grasping posture and do not consider the posture of the manipulator putting the object to the target position on the environment.

As for the research on pick-and-place planning of dual-arm manipulator includeing regrasping, Vahrenkamp et al.[4] proposed the inverse kinematics method and applied for the dual arm manipulator.

While this research obtains the planar area from the polygon models of the object and the environment, research has been done on finding planar area from point cloud data obtained using range sensors [17], [18], [19].

III. PICK-AND-PLACE PLANNING

The proposed planner plans the pick-and-place motion of dual-arm manipulators where the object is stably placed near the designated position on the environment. We especially consider putting the object at a concaved part or a planar part of the environment. The proposed planner is composed of offline and online phases. We first explain the offline phase where we model the surface of the object and the environment by using a set of planar regions. These regions allow us to speed up the calculation time of the online phase.

A. *Offline Surface Clustering*

Assuming that the surface of the object and the environment are modeled by polygons, we obtain a set of regions as a cluster of the triangular mesh. Let T_i $(i = 1, \cdots, m)$ be the i-th triangle of the object's surface model. Also, let C_j $(j = 1, \cdots, n)$ be the j-th cluster as a set of triangles on the object surface.

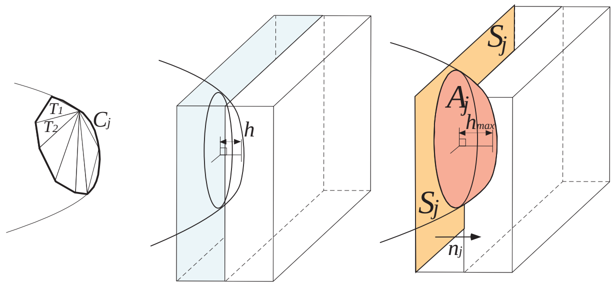
To obtain the region, we assume the flexibility at the contact surface. As shown in Fig.2(b), let h and h_{max} be the amount of deformation of the surface and its maximum value, respectively. Associated with the cluster C_j, let P_j be the regression plane fitted to the triangular mesh included in the cluster C_j. By using P_j, the contact plane S_j can be defined as a plane parallel to P_j where the distance between S_j and a tangent plane of the object surface is h_{max}. Also, let n_j be the unit outer normal vector of S_j.

We obtain a set of clusters on the object/environment surface under $h < h_{max}$. Although the clustering algorithm is different between the finger/object contact and the environment/object contact, the common part of the algorithm is summarized in Algorithm 1. In this algorithm, Step 1 computes the initial set of clusters where each is composed of a few triangles. Next, Step 2 calculates a cluster satisfying $h < h_{max}$. More concretely speaking, we first focus on a cluster C_x having the normal vector n_x of the contact plane P_x. Then, we consider merging a neighboring cluster where its normal vector is closest to n_x. By iterating this operation as far as the cluster C_x satisfies $h(C_x) < h_{max}$, we obtain the region on the surface of the object and the environment.

Algorithm 1 (Surface Clustering)

```
Step 1
$i \leftarrow 1$
for $j \leftarrow 1 : m$
    if $T_j \in C_j (j = 1 : i - 1)$, then continue
    $T_a \leftarrow SelectNeighborTriangle(T_j, n_j)$
    $C_i \leftarrow \{T_j, T_a\}$
    $i \leftarrow i + 1$
end
Step 2
$C_x \leftarrow SelectedCluster$
while $h(C_x) < h_{max}$
    $C_a \leftarrow SelectNeighborCluster(C_x, n_x)$
    $C_x \leftarrow \{C_x, C_a\}$
end
```

where $SelectNeighborCluster(C_x, n_x)$ denotes a function to select a cluster among the neighbors of C_x such that the direction of the normal vector is closest to n under $h(C_x) < h_{max}$.



(a) Definitions of triangles and clusters (b) Relation of contacting two bodies

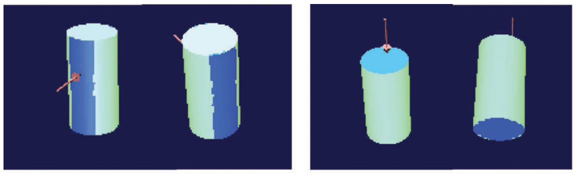
Fig. 2. Relation of contact between two bodies considering the flexibility of contact surface

1) Finger-Object Contact: In this subsection, we explain the offline clustering method of the object surface aiming to establish the contact between the object and the finger [15]. We assume a parallel gripper to grasp the object. To increase the grasp stability, we assume that a flexible sheet is attached at the finger surface.

We obtain multiple pairs of parallel contact planes so that the parallel gripper can stably enter in contact with the object. We first assign a point on the object surface. Then we execute Algorithm 1 to obtain the cluster C_x on the object surface. In addition to Algorithm 1, we further execute the clustering algorithm to obtain a cluster where the contact plane is parallel to that of C_x[15]. By assigning different part on the object surface, we iterate this algorithm and obtain multiple pairs of parallel contact planes on the object surface.

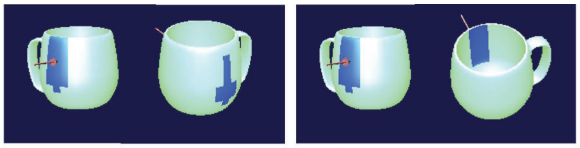
Once sets of parallel planes are computed, we obtain several parameters needed to plan the grasping posture[15]. Along with these parameters, the information on each cluster is described in a parameter file. When executing the online pick-and-place planner, we consider reading the parameter file so as to reduce the calculation time.

Figs.3 and 4 show some examples displaying the computed regions on the surface of the grasped object. For the polygon model of a drinking can, we obtained 4 pairs of regions on its surface where Fig.3 shows two of these. For the polygon model of a mug cup, we obtained 10 pairs of regions on its surface where Fig.4 shows 4 of these. Here, h_{max} is set as $0.005[m]$ in all examples.

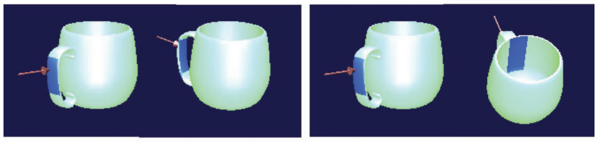


(a) A pair of cluster at the side of a can (b) A pair of cluster at the top/bottom of a can

Fig. 3. Two pairs of regions on the surface of a can



(a) Two pairs of clusters at the cup part



(b) Two pairs of clusters at the handle part

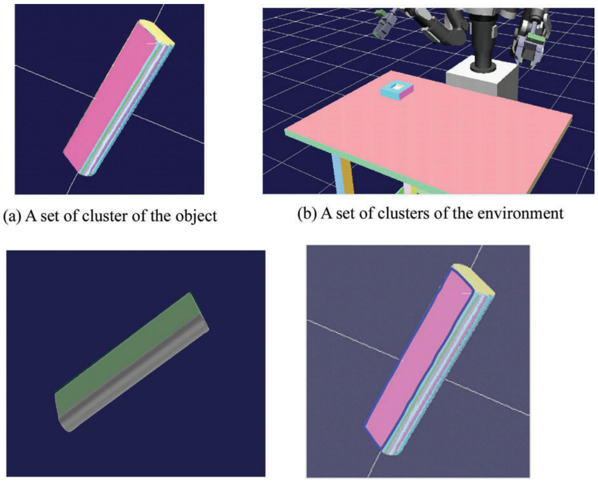
Fig. 4. Four pairs of regions on the surface of a mug cup

2) Object-Environment Contact: As well as the contact between the object and the finger, we obtain the region on the surface of the object and the environment so that the contact between the object and the environment can be established. Here, different from the contact between the finger and the object, there is no need to obtain a pair of contact planes parallel to each other on the object surface.

When we perform the online pick-and-place planning, we have to determine the relative position/orientation of the object to the environment and to judge whether or not the object is stably put on the environment.

For the polygon models of both the object and the environment, we iteratively execute Step 2 of Algorithm 1 by assigning different C_x defined in Step 1. The parameters of each cluster needed to plan the contact between the object and the environment are described in a parameter file. When executing the online grasp planner, we consider reading the parameter file.

Fig.5 shows an example of the regions on the surface of the object and the environment. In Fig.5(a), the colored areas show a set of clusters of the grasped object. Fig.5(b) shows the clusters of the environment model. Fig.5(c) and (d) shows the bounding box and the boundary of a cluster of the object model. Here, h_{max} is set as $0.001[m]$ in all examples.



(a) A set of cluster of the object (b) A set of clusters of the environment

(c) 2D bounding box of a cluster (d) Boundary of a cluster

Fig. 5. Clusters for the object/environment contact

B. Online Pick-and-Place Planner

The pick-and-place motion planner include several sub-planners; (1) grasp planner of the hand, (2) pose planner of the object placed on the environment, (3) manipulation strategy planner to select the manipulation strategy, and (4) trajectory planner of the robot to achieve the pick-and-place task.

1) Grasp Planner: We used the grasp planner proposed in the previous research[15]. Here, in addition to the previous research, we planned the grasping posture taking the key poses of the robot into consideration. The key poses include the pose of the robot approaching to the grasped object, grasping the object, lifting up the object, approaching to the placing point, and placing the object near the designated point on the environment. For given grasping posture of the object, we checked the inverse kinematics and the collision at each key pose. If we can find a set collision free key poses, we can find a collision-free pick-and-place motion trajectory for most of the cases.

2) Object Pose Planner: To stably put the object on the environment, we search for the position/orientation of the object placed on the environment. The user first assigns the final location of the object by selecting a cluster of the environment before executing the online pick-and-place planner. A cluster is selected by clicking a point on the environment on the graphics window. Then we search the clusters of the object contacting the environment. The overview of the algorithm is shown in the following:

Algorithm 2 (Search Object Posture Candidate)

```
2: for $j \leftarrow 1 : n$
3:     if $includeBoudingBox(C_j)$
4:         ObjectPostureSet ← searchFaceFaceContact
5:     else
6:         ObjectPostureSet ← searchEdgeFaceContact
7:     end
8:end
```

First, we check whether or not the surface of the object can be placed on a concaved part of the environment by comparing the edge length of the 2D bounding box (line 3). If 2D bounding box of the object is included in that of the environment, then we obtain candidates of the object posture assuming that the contact plane of the object contacting that of the environment (line 4). If it is not included, then we obtain candidates of the object posture assuming that a line forming the 2D convex hull of the cluster boundary contacts the contact plane of the environment (line 6). By using Fig.6, Algorithm 2 is detailed in the following:

$includeBoundingBox(C_j)$:

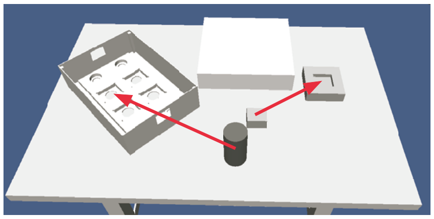
With this function, we compare the edge length of the 2D bounding box of a cluster. Fig. 6(b) shows an example where a can is placed on a dented part of a pallet. We compare the edge length of the 2D bounding box between the object and the environment. In this example, since the 2D bounding box of the object is included in that of the environment, the function $includeBoundingBox(C_j)$ returns TRUE. On the other hand, Fig. 6(d) shows an example where the function $includeBoundingBox(C_j)$ returns FALSE. We note that this function give a necessary condition for the object to put on a concaved part of the environment. Whether or not the object does not penetrate the environment is checked in the online grasp planner explained in the following subsection.

$searchFaceFaceContact$:

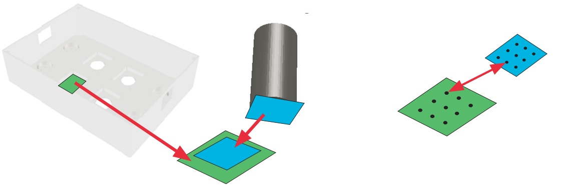
With this function, we obtain candidates of the object posture under the condition that a surface of the object maintains contact with a surface of the environment. As shown in Fig. 6(c), we set the contact plane of the object's cluster coincides with that of the environment's cluster. Under this situation we determine a position of the object placed on the environment. Then, we check whether or not the 2D bounding box of the object is included in that of the environment. If it is included, we check the stability condition shown in the following subsection. If the stability condition is satisfied, we append this position/orientation of the object to the $ObjectPostureSet$.

$searchEdgeFaceContact$:

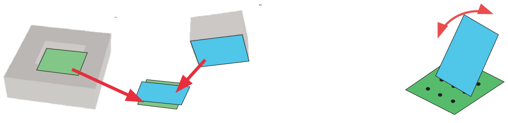
With this function, we obtain candidates of the object posture under the condition that the object maintains the line contact with one of the environment's clusters. As shown in Fig. 6(d), let us consider the situation where the 2D bounding box of the object is not included in that of the environment and where one of the edges of the object's bounding box is included in the 2D bounding box of the environment. In this case, we first calculate the 2D convex hull of the object's cluster boundary. Among the segments forming this convex hull, we set one of the segments contacting with the contact plane of the environment. Then we consider rotating the object about this segment until another contact between the object and the environment is established. If the contact is established, we append this position/orientation of the object to the $ObjectPostureSet$.



(a) Grasped object and environment



(b) Comparison of edge length of cluster bounding box (c) Generation of object posture candidates



(d) Comparison of edge length of cluster bounding box (e)Generation of object posture candidates

Fig. 6. Explanation of object placement planner

For the pose of the object where the object surface contacts the environment surface, we check whether the posture is statically stable. If we use only the bounding box of a cluster to check the stability, the object may fall into the hole since a bounding box does not include the information on the holes included in the object/environment surface.

We assume that the friction between the object and the environment is large enough and check if the vertical line including the CoG (center of gravity) of the object passes through the supported area. The overview of the algorithm is shown in Fig. 7. We first focus on the outer boundary of clusters and obtain the common area between the object cluster and the environment cluster. Then, we consider excluding the region of inner boundary from this common area. We further obtain the 2D convex hull of the area. Lastly, we check whether or not the vertical line including the object's CoG (center of gravity) passes through this area. If the area is included, we judge that the selected posture of the object is stable under the gravitational field.

3) Manipulation Strategy Planner: We prepared the following manipulation strategies: (1)*RIGHT-RIGHT:* Grasp the object using the right hand and put it using the right hand,

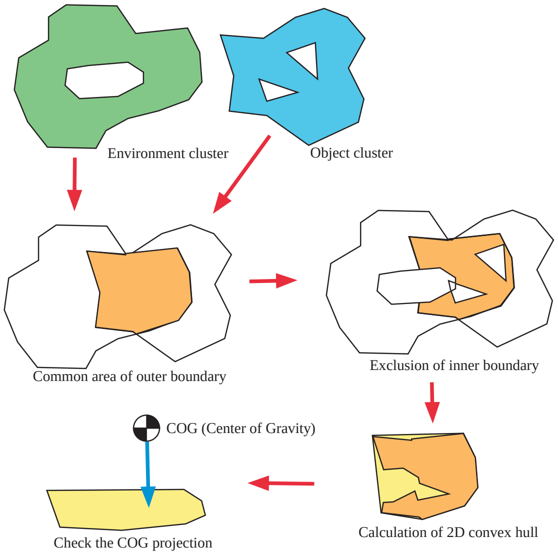


Fig. 7. Stability checking algorithm

(2)*LEFT-LEFT:* Grasp it using the left hand and put it using the left hand, (3)*RIGHT-LEFT:* Grasp it using the right hand and put it using the left hand by regrasping it, and (4)*LRFT-RIGHT* Grasp it using the left hand and put it using the right hand by regrasping it. For the initial and the target position of the object, we calculate which hand is closer to the object and determines the initial manipulation strategy. Here, depending on the object pose and on the collision between the arm and the environment, the pick-and-place planning may fail. In this case, we switch the manipulation strategy and plan the pick-and-place motion again.

C. Overall Planning Algorithm

Fig. 8 shows the online pick-and-place planning algorithm used in this research. We first generate the candidates of object placing posture by using the method explained in the subsection III-B.2. Then we select a manipulation strategy. Then, with selecting a object pose from its candidates, we plan the grasping posture of the object. After the grasping posture is obtained, we evaluate the grasping posture[15]. If it is judged that the obtained grasping posture has good quality, then we check the collision of the key postures included in the pick-and-place motion. By checking the collision, we can guarantee that the object does not penetrate the environment. If a series of the key posture are found out to be collision free, then we plan collision free trajectories between them by using the PRM (Probabilistic Roadmap Method).

IV. RESULTS

We confirmed the effectiveness of the proposed approach by simulation and experiment. We used the dual-arm manipulator NX-HIRO which has 2DOF head, two 6DOF arms, and 1DOF waist. Three cameras are attached at the head to measure the position/orientation of the object. We coded the pick-and-place planner as a plugin of the Choreonoid [20].

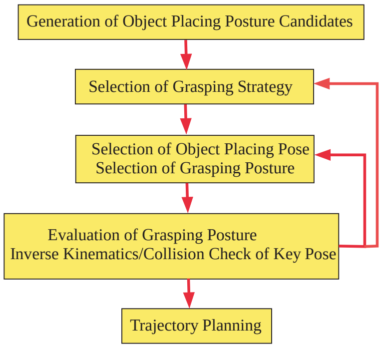


Fig. 8. Overall planning algorithm

Fig. 9 shows the result of calculation of the pick-and-place motion. We first clicked a dented part of the pallet on the graphics window. This part is shown by the arrow as shown in Fig. 9(a). Then we calculated the pick-and-place motion where the object is placed on the dented part of the pallet. Depending on the position of the object, the robot regrasps the object.

Fig. 10(a) shows the result where the area of the dented part is smaller than that of Fig. 9. In this case, just the edge of the object is placed on the dented part of the pallet. Fig. 10(b) shows the result of more complex shaped object where a mug cup is placed on a table surrounded by the obstacles.

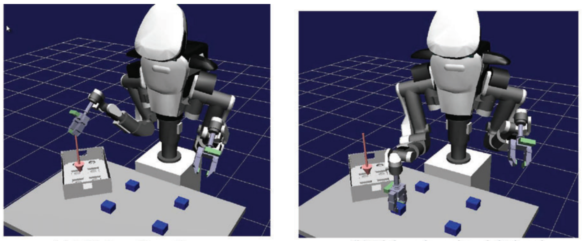
Fig. 11 shows the experimental result. The configurations of the robot, the object and the environment are same as the example of Fig. 9. In the experiment, we measured the position/orientation of the objects by using the stereo vision system based on edge-based segmentation[16].

V. CONCLUSIONS AND FUTURE WORKS

This paper proposed a method for planning the pick-and-place motion for dual-arm manipulators. Our planner is composed of the offline and the online phases. In the offline phase, we obtain a set of regions on the object/environment surfaces. In the online phase, we search for the grasping posture of the object, object pose placed on the environment, manipulation strategy, and the collision free trajectory of the robot. The effectiveness of the proposed approach was confirmed by simulation and experimental results by using the dual-arm robot NX-HIRO.

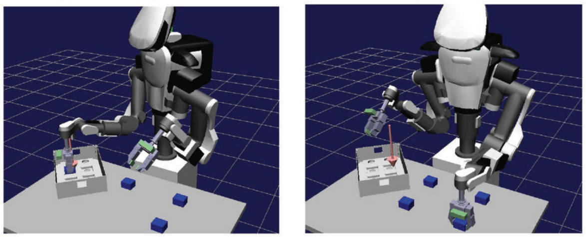
In this research, we consider putting the object at a concaved part or a planar part of the environment. The placement planner putting the object on any shape of the environment is considered to be a future research topic. Also, for the grasped object and the environment, it is assumed that the shape is known and the polygon model is prepared. Extension of the proposed method to the case where the shape of the object is not completely known is also considered to be a future research topic.

The authors would like express their sincere gratitude for Mr. Shunsuke Nozaki with AIST for his help in setting up the stereo vision. Lastly, this research was conducted under the support of the NEDO intelligent robot technology software project. Lastly, a part of the source code of the pick-and-place planner proposed in this research is available in [20].



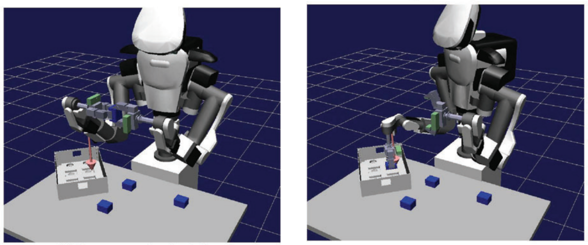
(a) Initial configuration

(b) Pick up by using right hand



(c) Put into the specified position on the pallet

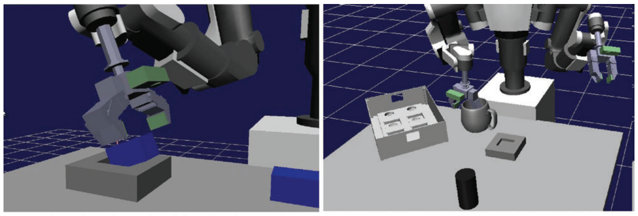
(d) Pick up by using left hand



(e) Regrasp using both hands

(f) Put into the specified position on the pallet

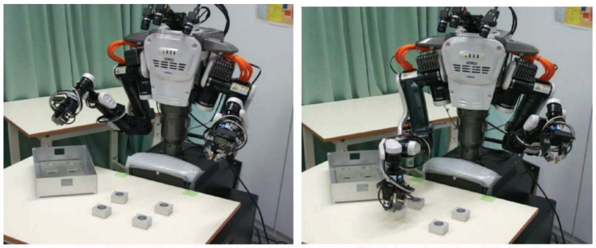
Fig. 9. Simulation result of the pick-and-place motion



(a) The robot places the object leaning on the environment

(b) The robot places the object among multiple obstacles

Fig. 10. Simulation result of putting object into a narrow space



(a) Initial configuration

(b) Pick up by using right hand

(c) Put into the specified position on the pallet

(d) Pick up by using left hand

(e) Regrasp using both hands

(f) Put into the specified position on the pallet

Fig. 11. Experimental result of the pick-and-place motion

REFERENCES

[1] C. Borst, M. Fischer, G. Hirzinger, “A fast and robust grasp planner for arbitrary 3D objects”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp.1890-1896, 1999.

[2] C. Borst, M. Fischer, G. Hirzinger, “Grasp Planning: How to Choose a Suitable Task Wrench Space”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp.319-325, 2004.

[3] A.T. Miller et al., “Automatic Grasp Planning using Shape Primitives”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 14-19, 2003.

[4] N. Vahrenkamp et al., “Humanoid Motion Planning for Dual-Arm Manipulation and Re-Grasping Tasks”, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2464-2470, 2009.

[5] D.C. Dupis et al., “Two-Fingered Grasp Planning for Randomized Bin-Picking”, *Proc. of the Robotics: Science and Systems 2008 Manipulation Workshop - Intelligence in Human Environments*, 2008.

[6] K. Nagata et al., “Two-Fingered Grasp Planning for Randomized Bin-Picking”, *Proc. of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.2109-2116, 2010.

[7] O. Ghita and P. F. Whelan, “A Bin Picking System Based on Depth from Defocus”, Machine Vision and Applications J., vol. 13, no. 4, pp. 234-244, 2003.

[8] K. Ikeuchi, “Generating an Interpretation Tree from a CAD Model for 3-D Object Recognition in Bin-Picking Tasks”, Int. J. Comput. Vision, vol. 1, no. 2, pp. 145-165, 1987.

[9] D. Berenson, J. Kuffner, and H. Choset, “An Optimization Approach to Planning Mobile Manipulation”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp.1187-1192, 2008.

[10] R. Diankov, “Automatid Construction of Robotic Manipulation Programs”, *Ph.D thesis, Carnegie Mellon University*, 2010.

[11] R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner, “BiSpace Planning: Concurrent Multi-Space Exploration”, *Proc. of Int. Symposium on Robotics, Science and Systems*, 2008.

[12] K. Harada, K. Kaneko, and F. Kanehiro, “Fast Grasp Planning for Hand/Arm Systems Based on Convex Model”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1162-1168, 2008.

[13] T. Tsuji, K. Harada, K. Kaneko, F. Kanehiro, and K. Maruyama, “Grasp Planning for a Multifingered Hand with a Humanoid Robot”, J. of Robotics and Mechatronics, Vol.22 No.2, pp. 230-238, 2010.

[14] T. Tsuji, K. Harada, and K. Kaneko, ”Easy and Fast Evaluation of Grasp Stability by using Ellipsoidal Approximation of Friction Cone”, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1830-1837、2009.

[15] K. Harada et al., “Grasp Planning for Parallel Grippers with Flexibility on its Grasping Surface”, *Proc. of IEEE Int. Conf. on Robotics and Biomimetics*, 2011.

[16] K. Maruyama, Y. Kawai, F. Tomita, ”Model-based 3D Object Localization Using Occluding Contours”, *Proc. of The Ninth Asian Conference on Computer Vision (9th ACCV)*, 2009.

[17] E. Klingbeil, A. Saxena, and A. Y. Ng, “Learning to Open New Doors”, *In Robotics Science and Systems (RSS) Workshop on Robot Manipulation*, 2008.

[18] PCL-Point Cloud Library, http://pointclouds.org

[19] N. Bergstrom, J. Bohg, and D. Kragic, “Integration of Visual Cues for Robotic Grasping”, *Proc. 7th Int. Conf. on Computer Vision Systems*, 2009.

[20] graspPlugin for Choreonoid, http://www.choreonoid.org/GraspPlugin/