2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Manipulation Motion Planner for Dual-Arm Industrial Manipulators

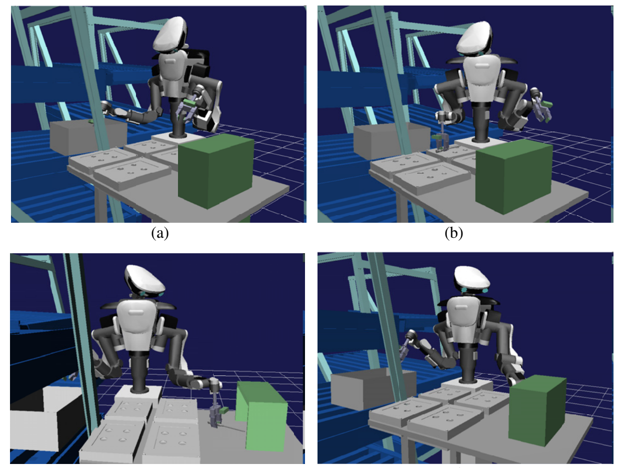
Kensuke Harada, Tokuo Tsuji, and Jean-Paul Laumond

***Abstract*— In this paper, we propose a general manipulation planner for dual-arm industrial manipulators. According to the context, the planner automatically determines whether both arms have to be used simultaneously or not. The approach is based on (i) the extension of an object placement algorithm previously developed in [23], and (ii) the introduction of several types of re-grasping motions dedicated to dual-arm manipulators. Such motions induce a special topological structure in the manipulation space that can be captured into a manipulation graph. The graph is then used to solve the manipulation problem by a simple graph search algorithm. After searching for a solution path, we further consider optimizing the path by minimizing the number of re-grasps. The effectiveness of the approach is demonstrated on the dual-arm manipulator HiroNX working in a realistic factory environment.**

I. INTRODUCTION

Object manipulation is one of the most typical tasks for a robot manipulator required to realize. In factory environments, dual-arm manipulators are widely used since they are expected to realize complex tasks including object manipulation. However, it is still difficult for a robot manipulator to automatically generate the motion of an object, especially when the manipulation task has to be performed in highly cluttered environment [1]. Fig. 1 shows an example of object manipulation by a dual-arm manipulator in a assembly cell. In this example, the robot picks up an object from a box and assembles it at a narrow area. Here, so as to assemble the object with an adequate grasping posture, the robot once has to place the object on the table by using the right hand and then re-grasp it by using the left hand. The paper addresses the manipulation problem in such a context. The contribution takes advantage of the geometric formulation of manipulation problems as introduced in [2], [4], [3] and further developed in various contexts [5], [6], [7], [8], [9], [10], [11], [12] to devise an original dual-arm planner capable to use both hands to manipulate an object. Both hands are used simultaneously or not according to the context, whereas the objective is to minimize the number of re-grasping. The paper also reports on the integration of the planner with a complete robot programming system including perception functions and on real-world experiments involving the robot HiroNX performing manipulation in realistic factory environments.

In [23], we have proposed planner allowing to automatically compute the stable placement of an object in a cluttered



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

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

environment. However, reaching such a placement from a given position may require several pick-and-place operations involving intermediate object placements, giving rise to a manipulation problem. On the other hand, previous works in manipulation planning do not integrate the automatic computation of the object placement space for cluttered environments. So, the manipulation planner we propose here may be considered as the integration of the object placement planner into an integrated manipulation scheme dedicated to dual-arm robots.

Dual-arm manipulators allow several types of re-grasping such as 1) re-grasping between the right and left hands, 2) once placing an object by using the right or left hand and then re-grasping it by using the right or left hand, and 3) once placing an object by using both hands and then re-grasping it by using both hands, etc. Moreover, dual-arm manipulator motions induce a special topology in the manipulation space. Indeed the manipulation space is structured into four foliated manifolds: the manifold in which the robot moves alone, the manifolds in which the left (respectively right) hand moves the object, and the manifold in which both hands move the object. A re-grasping strategy should be selected according to the context of the task. So as to seamlessly generate a re-grasping motion, we have to construct a manipulation graph that accounts for the special topology of the manipulation space of dual-arm manipulation. This graph is an extension of the graph proposed in [3] for a single manipulator. However both grasp and placement spaces are not explicitly given. Grasp and placement configurations are respectively

Kensuke Harada is with Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan kensuke.harada@aist.go.jp
Tokuo Tsuji is with the Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan tsuji@ait.kyushu-u.ac.jp
Jean-Paul Laumond is with LAAS-CNRS, Toulouse, France jpl@laas.fr

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

computed by the grasp planner proposed in [21], [22] and the object placement planner proposed in [23].

Since the generated manipulation paths may include many re-grasping motions, we furthermore propose an optimization method to reduce this number. The optimizer takes advantage of the foliation structure of the manipulation space to reduce the number of re-grasps.

To confirm the effectiveness of the proposed planner, we report on experiments involving the dual-arm industrial manipulator HiroNX working in a realistic and cluttered factory environment.

This paper is organized as follows: after introducing the related works in Section 2, Section 3 introduces the definitions used in the paper. We explain the detail of the manipulation planner in Section 4. Section 5 describes the path optimizer. In Section 6, we show the effectiveness of the proposed method through simulation examples and real experiments.

II. RELATED WORKS

Recently, there has been a lot of researches on manipulation planning. As for a single arm manipulation, Vahrenkamp et al.[7] proposed the methods to combine the grasp and the path planners. Hauser et al. [17] proposed a method for multi-step motion planning problem. Berenson et al. [9] proposed a manipulation planner between different sub-manifold of configuration space. Simeon et al. [3] proposed a manipulation planner for a single manipulator by considering a series of re-grasping and placing motions. Some researchers proposed a method to grasp an object after pushing it on the table [5], [6].

As for the dual arm manipulation [18], Koga et al. [4] proposed a framework of manipulation planning for multiple arm systems. Gharbi et al. [16] proposed a two-step method for roadmap composition for multi-arm systems. Saut et al. [8] proposed a re-grasping planner between the left and the right hands.

However, there has been no research on manipulation planning for dual-arm manipulators where, by integrating the grasp and the object placement planners, the function of using both hands or not according to the context is realized. Also, there has been no path optimizer minimizing the number of re-grasping.

III. DEFINITIONS

In this section, we introduce the definitions required to describe a manipulation problem.

A. Notation

Fig.2 shows the manipulation space used to plan a manipulation task. Let us consider a robot having n-DOF right arm and n-DOF left arm. Let CS_r, CS_l and CS_o be the configuration space of the right arm, the left arm and the object, respectively. The manipulation space is defined by $CS = CS_r \times CS_l \times CS_o$. Let CS_{free} be the collision-free subset of CS. Let CP be the domain in CS where the object is stably placed on the environment. Also, let CG_r and CG_l be the domain in CS where the object is stably grasped by the right and the left hands, respectively. CP, CG_r and CG_l are sub-dimensional manifolds in CS.

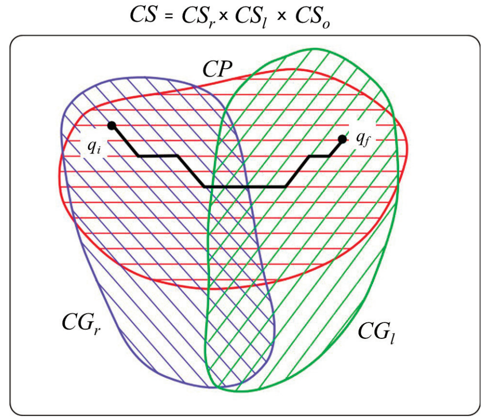


Fig. 2. The manipulation space gathers the three configuration spaces of the object, the left arm and the right arm respectively.

B. Grasp Planner

A stable grasp is realized by using the grasp planner proposed in [21], [22]. Before executing the manipulation planner, the grasp planner generates multiple candidates of stable grasping postures for a given object. More concretely speaking, the output of the grasp planner is a set of position/orientation of the palm with respect to the object coordinate system and the finger joint angles. We define the i-th candidate of the right-hand grasp g_{ri} and the left-hand grasp g_{li} in CG_r and CG_l, respectively.

C. Object Placement Planner

Stable object placements are computed by using the object placement planner [23]. The object placement planner generates multiple object postures stably placed on the environment surface. We denote by p_j the j-th object placement in CP. A combination of the right-hand grasp g_{ri} and the placement p_j defines a configuration in $CG_r \cap CP$.

D. Manipulation Paths

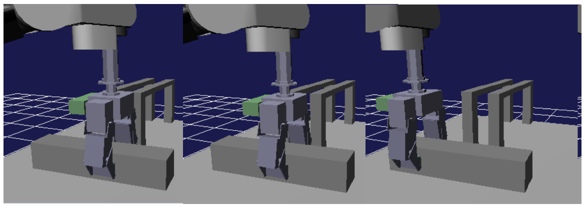
A solution path of the manipulation planning problem consists of the following two kinds of paths [3] (Fig.3):

Transit Path where the robot moves alone while the object stays stationary in a stable position. Transit paths induce a foliation of CP.

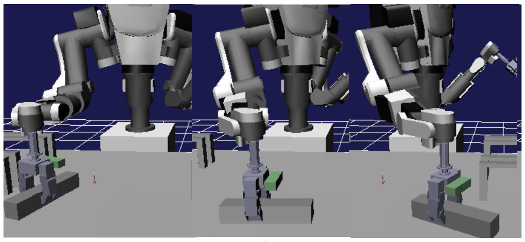
Transfer Path where the robot moves while stably grasping the object. Transfer paths induce a foliation of CG_r or CG_l.

The manipulation planning problem is to find a sequence of the transfer and the transit paths connecting the starting configuration q_i and the goal configuration q_f in $CP \cup CG_r \cup CG_l$.

We now focus on $CG_r \cap CP$. Two foliation structures are induced respectively by the transfer paths and by the transit paths. Let us consider two configurations corresponding to



(a) Transit path



(b) Tranfer path

Fig. 3. Transfer and tansit paths

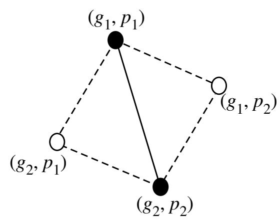


Fig. 4. Three types of node connection

the grasp g_{r1} and g_{r2} and the object placement p_1 and p_2, noted $(g_{ri}, p_i)_{i=1,2}$. To connect (g_{r1}, p_1) to (g_{r2}, p_2), we can define two intermediate nodes as (g_{r1}, p_2) and (g_{r2}, p_1). As shown in Fig.4, now we can define three kinds of paths to connect (g_{r1}, p_1) to (g_{r2}, p_2):

- **Type1**: A direct path from (g_{r1}, p_1) to (g_{r2}, p_2) lying inside $CG_r \cap CP$.
- **Type2a**: A transfer path from (g_{r1}, p_1) to (g_{r1}, p_2) followed by a transit path from (g_{r1}, p_2) to (g_{r2}, p_2).
- **Type2b**: A transit path from (g_{r1}, p_1) to (g_{r2}, p_1) followed by a transfer path from (g_{r2}, p_1) to (g_{r2}, p_2).

Type1 paths are not physically realizable. If two nodes are connected by using a Type1 path, it is replaced by a sequence of Type2a or Type2b paths in a post process of the manipulation plan. This is the consequence of the so-called reduction property demonstrated in [3]. We also note that, although we just focus on $CG_r \cap CP$, three kinds of paths can also be defined in $CG_l \cap CP$.

On the other hand, $CG_r \cap CG_l$ also has foliation structure. Especially, $CG_r \cap CG_l \cap CP$ has three-foliation structure. However, as shown in the next section, we can plan the object manipulation without considering the foliation structure in $CG_r \cap CG_l$.

IV. MANIPULATION PLANNER

A. Graph Structure

Based on the manipulation space defined in the previous section (Fig.2), we now construct the manipulation graph. First, we consider categorizing the object placement CP into three classes: initial, intermediate and target placements, respectively. Then, according to the grasping state CG_r, CG_l and $CG_r \cap CG_l$, the manipulation space is divided into 10 components as shown in Fig. 5. Here, each component includes a set of configurations denoted as nodes.

The component belonging to the initial object placement has a single object pose with multiple grasping postures since the grasp planner generates multiple grasping configurations. On the other hand, in the intermediate object placement, multiple object configurations may be associated to multiple grasping configurations.

Two components are connected by using one of the three kinds of paths: transit, transfer or Type2 path. Also, although all components of the manipulation graph has a foliation structure, the search is performed only by considering the foliation structures of the intersection between the intermediate object placement and the right-hand and left-hand hand configuration spaces. The reason will be explained later in this section.

B. Graph Components

We now explain how to generate each component of the manipulation graph.

1) Initial Object Placement: For the initial object placement p_{ini}, we consider the right-hand grasp g_{ri} and the left-hand grasp g_{li}, $(i = 1, \cdots, I)$. Let us consider the case of $CG_r \cap CP$, we consider solving the inverse kinematics of the right arm for given combination of (g_{ri}, p_{ini}). If the inverse kinematics is solvable and the arm posture is collision free, we consider applying this configuration as a node included in the component of the right-hand grasping space at the initial placement. The same discussion applies for $CG_l \cap CP$ and $CG_r \cap CG_l \cap CP$.

2) Intermediate Object Placement: $CG_r \cap CP$ and $CG_l \cap CP$ manifolds are structured along two foliations. So any path in these manifolds may be approximated by a sequence of manipulation paths. We make use of the Visibility-PRM algorithm [13], [3] to capture the connected components of these manifolds. Visibility-PRM tends to generate a minimum number of nodes covering $CG_r \cap CP$ and $CG_l \cap CP$. The steering method used to connect two nodes is based on the computation of Type1 paths. By using the Visibility-PRM, we can expect that the number of re-grasps included in the solution path is not large.

Let us now consider $CG_r \cap CG_l$. For simplicity, we do not have to consider the foliation structure of this component. We consider generating a set of configurations in $CG_r \cap CG_l$ by randomly sampling the object pose in 3D and the grasping pose of each hand.

3) Target Object Placement: Let us consider the object configuration at the target. For a set of target object placement p_{fj} $(j = 1, \cdots, J)$, we consider a set of grasp g_{ri} and g_{li}, $(i = 1, \cdots, I)$. Within $CG_r \cap CP$, we consider solving the inverse kinematics of the right arm for given (g_{ri}, p_{fi}). If the inverse kinematics is solvable and the arm configuration is collision-free, we consider applying this configuration as a node included in this component of the graph. The same discussion applies for $CG_l \cap CP$ and $CG_r \cap CG_l \cap CP$.

C. Component Connection

Now, we consider connecting the various components of the graph.

1) Type2 Path: Let us consider the case where two components are connected by using a Type2 path. We consider randomly selecting one node from both components and trying to connect them by using a Type2 path. We consider iterating this operation for a predefined times.

2) Transfer Path: To connect two components by using a transfer path, we first randomly select a node from one of the components. Then, with keeping the same grasp, we consider changing the robot posture so that the configuration is included in the other component. If we can find such a collision-free path, we conclude that the connection between two components can be established by using a transfer path.

3) Transit Path: To connect two components by using a transit path, we first randomly select a node from one of the components. Then, with keeping the same object placement, we consider changing the grasp so that the configuration is included in the other component. If we can find such a collision-free path, we conclude that the connection between two components can be established by using a transit path.

4) MPK(Motion Planning Kit): In this work, two components of the graph are connected by using either Type2, transit or transfer paths. For all cases, a path between two configurations is computed by using MPK motion planner (Motion Planning Kit) [24]. MPK first tries to connect two configurations by using a straight line path. If the connection is not collision-free, then MPK tries to connect two configurations by using the single-query, bi-directional, and lazy-collision checking planner[15].

5) Path Generation: Let us assign a group ID for each node. When connecting two nodes, we set that the ID is the same for two nodes. After connecting two components included in the manipulation graph, we check the ID of initial and target nodes. If the IDs are the same for both nodes, we search for the path connecting them. We use the depth-first search to obtain a solution path. If Type1 path is included in the solution path, we consider replacing it by using a Type2 path.

Realizing a task by a dual-arm manipulator may require multiple manipulation styles. For example, pick-and-place in a simple environment may be realized only by using the right arm. The same task may be realized only by using the left arm. Or, sometimes the same task may be realized by once picking up the object with the right arm, and then placing it with the left arm. The manipulation style selected by using our proposed planner depends on the order of the component connection. Here, depending on the task requirements, one of the manipulation styles cannot be executed. In this case, preparing multiple manipulation styles becomes effective. Here, the optimization of the manipulation style is considered to be our future research topic.

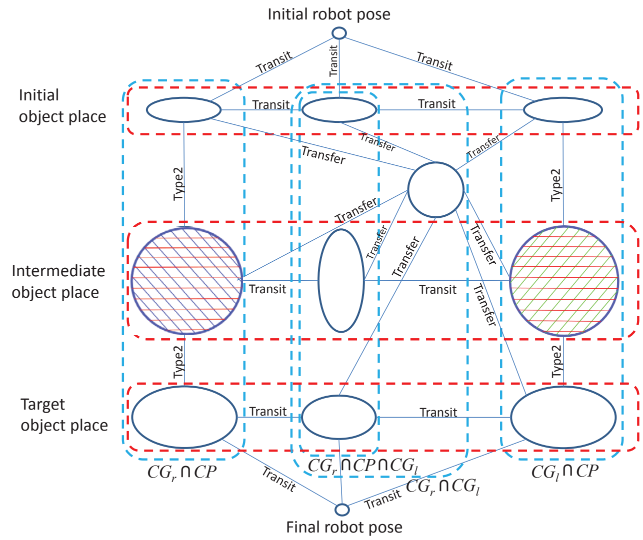


Fig. 5. Graph structure used in manipulation planning

V. PATH OPTIMIZATION

The manipulation path obtained in the previous section may include several re-grasping and placement operations. To reduce the motion time and increase the reliability of manipulation, it is desirable that the number of re-grasps is minimized. This section discusses a method for optimizing the manipulation path by reducing the number of re-grasp/placement operations by using a single arm. We propose three operations as shown in Fig.6.

1) Transfer-Transfer: If a transfer path comes after another transfer path, we consider merging both transfer paths. We try to connect the initial configuration of the former path with the final configuration of the latter path by using a transfer path. If connection is established, we consider applying this shortcut path.

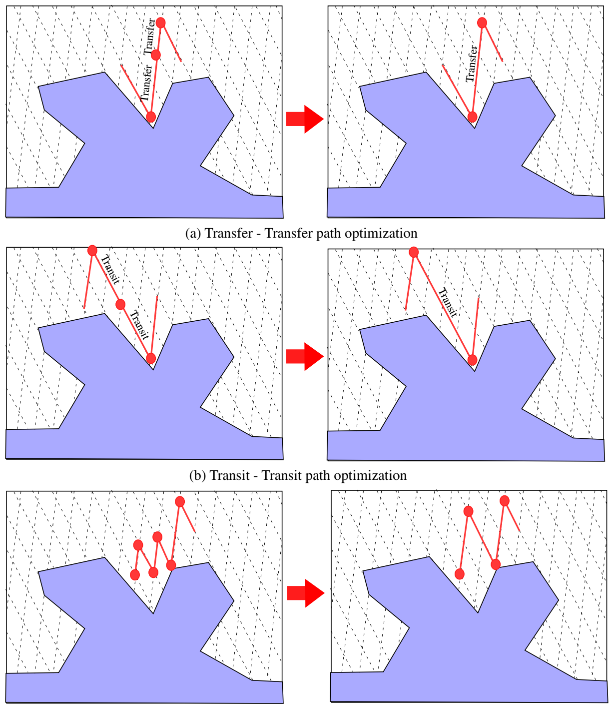
2) Transit-Transit: If a transit path comes after another transit path, we consider merging both transit paths. We try to connect the initial configuration of the former path with the final configuration of the latter path by using a transit path. If connection is established, we consider applying this shortcut path.

3) Type2-Type2: If a Type2a path comes after another Type2a path, we consider merging both Type2a paths. Also, if a Type2b path comes after another Type2b path, we consider merging both Type2b paths.

We iteratively execute the above three operations for a predefined times.

VI. EXAMPLES

To confirm the effectiveness of the proposed manipulation planner, we performed several simulation examples and experiments. The planner was applied to the robot HiroNX developed by Kawada Industries Inc. HiroNX has two 6-dof arms and two two-fingered hands. The planner was coded on the graspPlugin for Choreonoid [25]. In the graphics window, the user clicks the target position of the

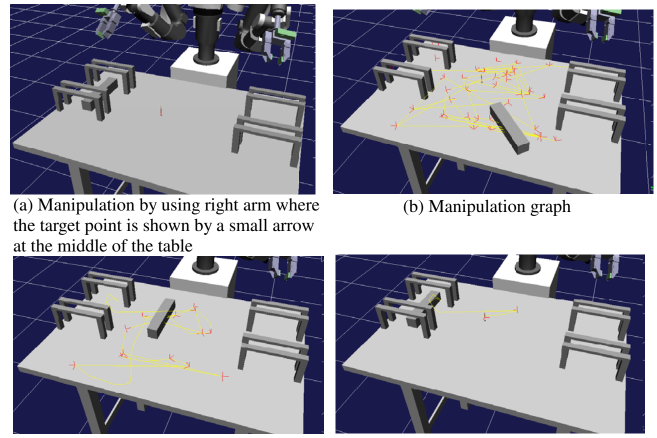


(a) Transfer - Transfer path optimization

(b) Transit - Transit path optimization

(c) Type2 - Type2 path optimization

Fig. 6. Three path optimization methods



(a) Manipulation by using right arm where the target point is shown by a small arrow at the middle of the table

(b) Manipulation graph

(c) Solution path

(d) Optimized path

Fig. 7. Manipulation by using right arm

object in the environment. Then, the manipulation planner computes a solution path (if any[1]) and the robot performs the pick-and-place operations to place the object at the target position. Several video of the experiments can be seen on http://staff.aist.go.jp/kensuke.harada/ManipPlanner.

First, we run the manipulation planner in the environment as shown in Fig.7. We consider moving a rectangular object from a cage and place it at the middle of the table. Generated manipulation graph is shown in Fig.7(b). The solution path and its optimization are shown in Fig.7(c) and (d), respectively. As shown in this figure, the number of re-grasps is much reduced. Fig.8 shows the motion of the robot moving the object from the initial to the final placement. The solution path in the manipulation graph (Fig.5) is shown in Fig. 10(a).

In the second example, we also consider placing the object at the middle of the cage located at the left-hand side of the robot. Although it is possible for the robot to grasp and to place the object by using the right hand, we first connected the component of $CG_r \cap CP$ at the intermediate placement with the component of $CG_r \cap CG_l \cap CP$ at the same placement and connected the component of $CG_l \cap CP$ at the intermediate placement with the component of $CG_r \cap CG_l \cap CP$ at the same placement. Then, the robot first grasps the object by using the right hand and place it at the middle of the table. Then, the robot re-grasps the object by using the left hand and places it at the desired placement. The solution path in the manipulation graph is shown in Fig. 10(c).

We also performed real world experiments. The experimental setup is shown in Fig.11. The robot is placed in the physical mock-up of a cell production system. In this environment, the robot tries to pick up an object located in a tray and put it in a box placed on a table. As shown in the figure, the 3D data of environment is captured by using a Kinect sensor and converted into a polyhedral model. The figure also shows the 3D model of the grasped object.

Fig.12 shows an experimental result of pick and place by using the right arm. On the other hand, Fig.13 shows an experimental result of re-grasping between the right and left hands. In this experiment, the robot is ordered to place the object on the box with upside-down posture. Since the re-grasping is needed for this example, the re-grasping motion is selected. In both examples, the robot can stably place the object while the environment is highly cluttered. The solution path in the manipulation graph (Fig.5) is shown in Fig. 10(b).

Lastly, the calculation time of all examples is between 1[m] and 2[m]. Currently, the calculation time is not short since, for many components of the manipulation graph, a set of collision free configurations are found by using random sampling. Reducing the calculation time is considered to be our future research topic.

VII. CONCLUSIONS

This paper proposed a dual arm manipulation planner that can realize several manipulation styles including pick-and-place operations. The proposed planner considers the sub-dimensional manifolds in the so-called manipulation space. We showed through examples that several manipulation styles may be considered, such as single arm grasping, regrasping from the one to the other hand, regrasping and placing, and bimanual grasping.

In this paper, we have not considered the motion path involving the closed kinematic chain imposed by bimanual manipulation [26]. With our current setting, although we can realize the bimanual grasp, we cannot manipulate the object with grasping two hands. Implementation of the path planner with closed kinematic chain [26] is considered to be our future research topic. Also, search for the optimal

[1]Notice that the planner is probabilistically complete, i.e. it may not find a solution whereas such a solution exists.

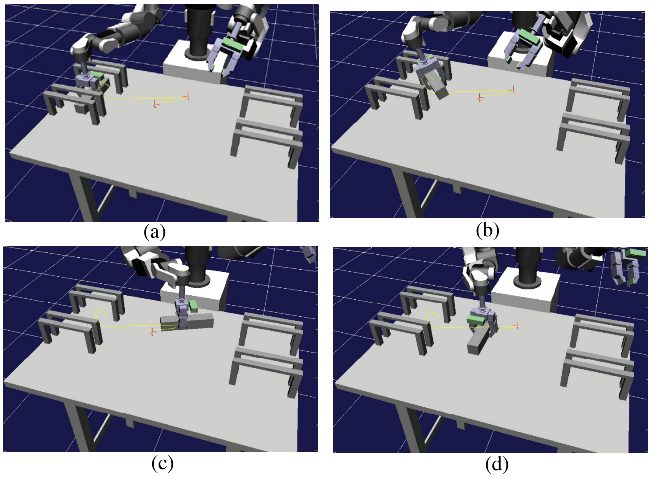


Fig. 8. Motion of robot manipulating an object by using right arm

manipulation style is considered to be our future research topic.

Acknowledgements: The authors would like to express their sincere graditude for the members of the vision and manipulation research group of AIST and the members of the Gepetto team of LAAS-CNRS for their helpful discussions.

References

[1] "New Generation Robot", *Yaskawa Technical Review*, vol. 72, no. 2, 2008.

[2] B. Dacre Wright, J.P. Laumond, R. Alami, "Motion Planning for a Robot and a Movable Object Amidst Polygonal Obstacles", *IEEE International Conference on Robotics and Automation*, pp. 2474-2480, 1992.

[3] T. Simeon, J.P. Laumond, J. Cortes, and A. Sahbani, "Manipulation Planning with Probabilistic Readmaps", Int. J. Robotics Research, vol. 23, no. 8, pp. 729-746, 2004.

[4] Y.Koga and J.C. Latombe, "On Multi-arm Manipulation Planning", *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1994.

[5] L.Y. Chang, S.S. Srinivasa, and N.S. Pollard, "Planning pre-grasp manipulation for transport tasks", *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2010.

[6] M.R. Dogar and S.S. Srinivasa, "A Framework for Push-grasping in Clutter", Robotics: Science and Systems VII, 2011.

[7] N. Vahrenkamp, T. Asfour, R. Dillmann, "Simultaneous Grasp and Motion Planning", Robotics and Automation Magazine, 2012.

[8] J.-P. Saut, M. Gharbi, J. Cortes, D. Sidobre, T. Simeon, "Planning Pick-and-Place tasks with two-hand regrasping", *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.4528-4533, 2010

[9] D. Berenson, S.S. Srinivasa, D.Ferguson, and J.J. Kuffner, "Manipulation Planning on Constraint Manifolds", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 625-632, 2009.

[10] D. Berenson, S.S. Srinivasa, D. Ferguson, A. Collety, and J.J. Kuffner, "Manipulation Planning with Workspace Goal Regions", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 618-624, 2009.

[11] N. Vahrenkamp, D. Berenson, T. Asfour, J.J. Kuffner, R. Dillmann, "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.

[12] 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.

[13] T. Simeon, J.P. Laumond, and C. Nissoux, "Visibility-based Probabilistic Roadmaps for Motion Planning", Advanced Robotics, vol. 14, no. 6, 2000.

[14] J.P. Laumond and T. Simeon, "Notes on Visibility Roadmaps for Motion Planning", *Algorithmic Foundations of Robotics(WAFR 00)*, 2001.

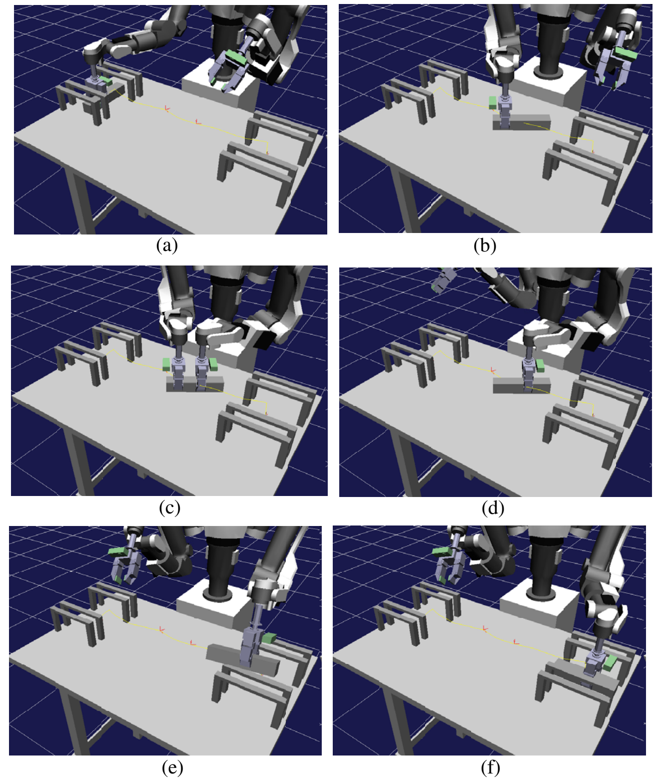


Fig. 9. Motion of the robot manipulating the object with regrasping/placing

[15] G. Sanchez and J.C. Latombe, "A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking", *Int. Symposium on Robotics Research (ISRR'01)*, 2001.

[16] M. Gharbi, J. Cortes, and T. Simeon, "Roadmap Composition for Multi-Arm Systems Path Planning", *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009

[17] K. Hauser and J.-C. Latombe, "Multi-modal Motion Planning in Non-expansive Spaces", int. J. Robotics Research, 2009.

[18] C.Smith et al., "Dual arm manipulation–A survey", Robotics and Autonomous Systems, vol. 60 pp.1340-1353, 2012.

[19] 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.

[20] 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.

[21] 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.

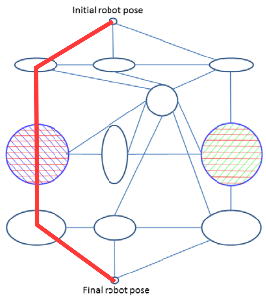
[22] K. Harada et al., "Pick and Place Planning for Dual-Arm Manipulators", *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2281-2286, 2012.

[23] K. Harada et al., "Object Placement Planner for Robotic Pick and Place Tasks", *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 980-985, 2012.

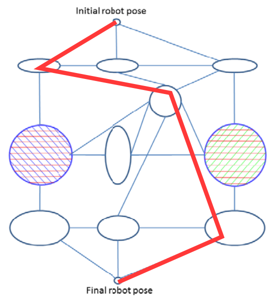
[24] MPK(Motion Planning Kit), http://robotics.stanford.edu/m̃itul/mpk

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

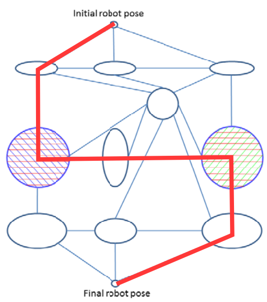
[26] J. Cortes, T. Simeon, and J.P. Laumond, "A Random Loop Generator for Planning the Motions of Closed Kinematic Chains with PRM Methods", *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2002.



(a) Manipulation by using right arm

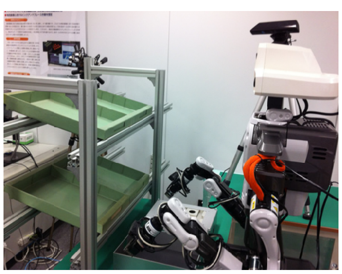


(b) Manipulation with regrasping

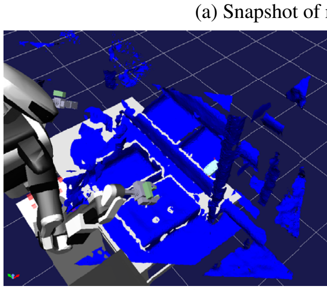


(c) Manipulation with regrasping/placing

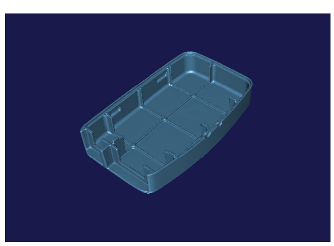
Fig. 10. Solution path in the manipulation graph



(a) Snapshot of real environment

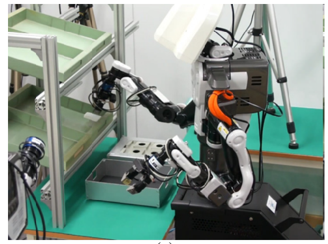


(b) Polygon model of environment



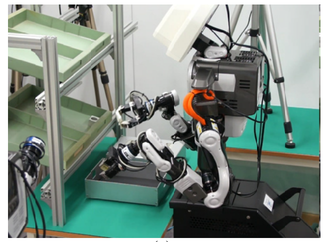
(c) Polygon model of object

Fig. 11. Experimental Setup



(a)

(b)



(c)

(d)

Fig. 12. Experimental Result

(a)

(b)

(c)

(d)

(e)

(f)

Fig. 13. Experimental Result