Theory and Implementation of Dual-Arm Manipulation Planning

Kensuke Harada

***Abstract*— In this paper, we study theory and implementation issues of dual-arm manipulation planning. We mainly discuss on three topics: (1) static geometrical approach on planning the position/orientation an object placed at the designated part on the environment, (2) pick-and-place planning of a dual-arm manipulator where the planner automatically determines whether both arms have to be used simultaneously or not according to the context, and (3) implementation issues of dual-arm manipulation where manipulation planning can be performed within a reasonable time. The effectiveness will be verified through several experimental results.**

I. INTRODUCTION

In factory environments, dual-arm manipulators are widely used since they are expected to realize complex tasks including object manipulation[1], [2]. The paper addresses some topics existing in the dual-arm manipulation planning problem.

We firstly discuss the static geometrical approach on object placement planner[3] where the planner automatically determines the pose of an object that is stably placed near a user-assigned point on the environment surface. In this method, the polygon models of both the object and the environment are clustered, with each cluster being approximated by a planar region.

We secondly discuss the pick-and-place motion planner[4] where the planner automatically determines whether both arms have to be used simultaneously or not according to the context. We can consider several types of re-grasping by a dual-arm manipulator 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. By utilizing a special topological structure in the manipulation space that can be captured into a manipulation graph, the manipulation problem is solved by searching a manipulation graph.

We thirdly consider the problem solving the pick-and-place planning within a reasonable time. We consider a case where the shape complexity of the fixture highly restricts the posture of a robot placing an object to the fixture. In our method, we consider simultaneously generating three postures of the robot: 1) picking up an object, 2) regrasping an object from the right hand to the left hand, and 3) placing an object to the fixture. Our planning algorithm effectively utilizes multiple threads and reduces the size of the search space.

Kensuke Harada is with Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan `kensuke.harada@aist.go.jp`

II. OBJECT PLACEMENT PLANNER

Our object placement planner is composed of an offline and an online phases. In the offline phase, we find planar clusters on both the object and the environment surfaces by clustering the polygon model. We then consider assigning some properties such as **Contacting Cluster** and **Convexity** to each cluster. In the online phase, to check whether or not a planar cluster of an object can enter in contact with a planar cluster of an environment, we first introduce two testing methods (**Convexity Test** and **Contact Test**). **Convexity Test** is introduced for run-time efficiency of the online searching method. By checking the convexity/concavity relation between the object and the environment, **Convexity Test** considers reducing the number of candidate clusters where the collision is checked between the object and the environment. Then, **Contact Test** checks collision between the object and the environment for given position/orientation of the object placed on the environment. After both testing methods are passed, we furthermore check whether or not the object can keep the gravitational equilibrium placed on the environment surface (**Stability Test**).

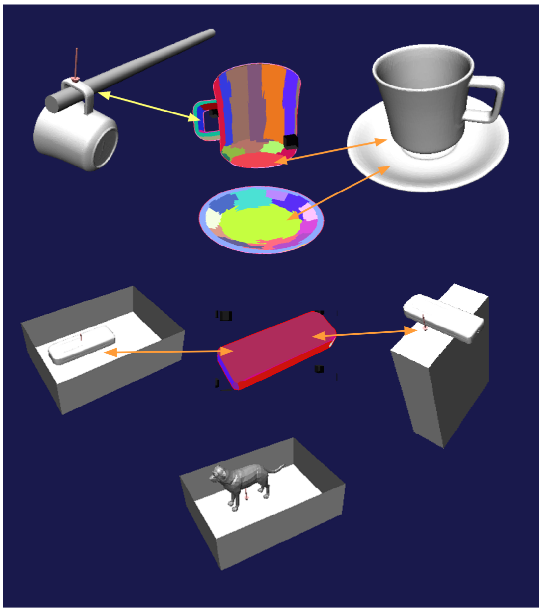

Fig. 1. Example of object pose obtained by using the object placement planner

III. DUAL-ARM MANIPULATION PLANNER

Fig.2 shows the manipulation space used to plan a manipulation task. Let CS_r, CS_l and CS_o be the configura-

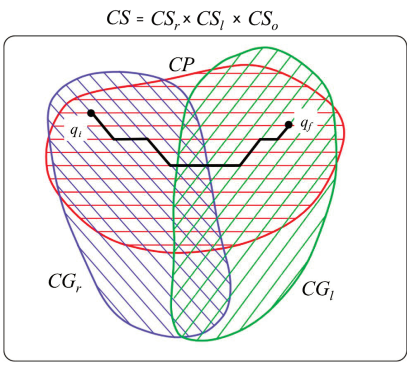

Fig. 2. The manipulation space gathers the three configuration spaces of the object, the left arm and the right arm respectively.

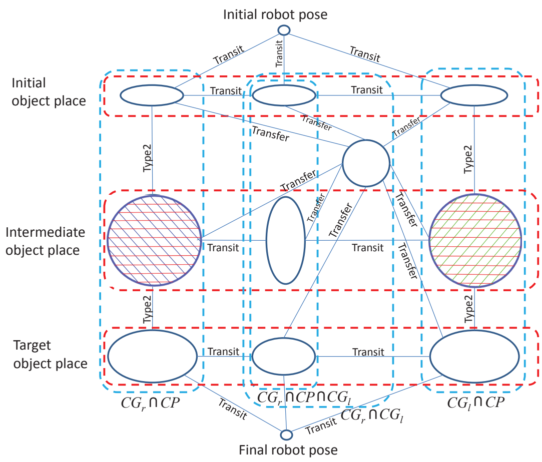

Fig. 3. Graph structure used in manipulation planning

tion space of the right arm, the left arm and the object, respectively. Let CG_r and CG_l be the domain in CS where the object is stably grasped by the right arm and the left arm, respectively. Let CP be the domain in CS where the object is stably placed on the environment. Each subspace has the folilation structure according to the stable grasp and the stable placement.

Based on the manipulation space, the manipulation graph is constructed as shown in Fig. 3. By searching for this manipulation graph, we can seamlessly generate pick-and-place task including several styles of regrasping as mentioned in the introduction.

IV. IMPLEMENTATION OF DUAL-ARM MANIPULATION

This section especially focuses on the manipulation planning where a dual-arm manipulator first picks up an object from the pile, then regrasps it from the right hand to the left hand, and finally places it to the fixture. We consider simultaneously generating three postures of the robot: the grasping posture, the regrasping posture and the placing posture. Among three postures, calculation of the regrasping posture is the most time consuming. Hence, to plan the pick-and-place motion, we have to make the size of candidate grasping and placing pose as small as possible before searching the regrasping posture. Also, the geometrical complexity of the fixture highly restricts the placing posture of a robot, the searching space for finding robot pose becomes large. In this case, we have to solve the inverse kinematics for a number of times during the searching process. Hence, we constructed a function of solving the inverse kinematics problem in parallel. Let $DG_r(obj)$ be the database of right-hand's grasping posture for the object named obj. For given pose of an object P_o, let the function rightArm.parallelIK $(P_o, DB_r(obj))$ output the subset of $DG_r(obj)$ where the inverse kinematics problem is solvable. Let P_{oi} and P_{of} be the initial and the target pose of the object. When regrasping an object, we consider selecting an object pose from the set $DP(obj)$. The pseudo code of our planning problem becomes as follows:

Algorithm 1 (Calculation of Robot Postures)

1: $DG'_r(obj) \leftarrow$ rightArm.parallelIK$(P_{oi}, DG_r(obj))$
2: $DG''_r(obj) \leftarrow$ rightArm.checkCollision$(P_{oi}, DG'_r(obj))$
3: $DG'_l(obj) \leftarrow$ leftArm.parallelIK$(P_{of}, DG_l(obj))$
4: $DG''_l(obj) \leftarrow$ leftArm.checkCollision$(P_{of}, DG'_l(obj))$
5: for $i \leftarrow 1 : DG''_r(obj)$.size()
6:   for $j \leftarrow 1 : DG''_l(obj)$.size()
7:     $DP'(obj) \leftarrow$ rightArm.parallelIK$(DP(obj), DG''_r(obj)[i])$
8:     $DP''(obj) \leftarrow$ leftArm.parallelIK$(DP'(obj), DG''_l(obj)[j])$
9:     for $k \leftarrow 1 : DP''(obj)$.size()
10:       if not rightArm.IK$(DP''(obj)[k], DG''_r(obj)[i])$, continue
11:       if rightArm.Colliding$(DP''(obj)[k], DG''_r(obj)[i])$, continue
12:       if not leftArm.IK$(DP''(obj)[k], DG''_l(obj)[j])$, continue
13:       if leftArm.Colliding$(DP''(obj)[k], DG''_l(obj)[j])$, continue
14:       return $DP''(obj)[k]$, $DB''_r(obj)[i]$, $DB''_l(obj)[j]$
15:     end
16:   end
17: end

Fig. 4 shows an experimental result where a dual-arm manipulator first picks up an object from the pile, regrasps it from the right to the left hand, and places it to the fixture by using the left hand.

REFERENCES

[1] "New Generation Robot", *Yaskawa Technical Review*, vol. 72, no. 2, 2008.
[2] C.Smith et al., "Dual arm manipulation–A survey", Robotics and Autonomous Systems, vol. 60 pp.1340-1353, 2012.
[3] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, and H. Onda, "Validating an Object Placement Planner for Robotic Pick-and-Place Tasks", J. of Robotics and Autonomous Systems, vol. 62, pp. 1463-1477, 2014.
[4] K. Harada, T. Tsuji, and J.-P. Laumond, "A Manipulation Motion Planner for Dual-Arm Industrial Manipulators", *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 928-934, 2014.
[5] K. Harada et al., "Project on Development of a Robot System for Random Picking –Grasp/Manipulation Planner for a Dual-arm Manipulator–", *Proc. of IEEE/SICE Int. Symp. on System Integration*, pp. 583-589, 2014.
[6] K. Harada, T. Tsuji, K. Kikuchi, K. Nagata, H. Onda, and Y. Kawai, "Planning a Sequence of Pick-and-Place Tasks using Dual-arm Mobile Manipulators", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015. (submission)

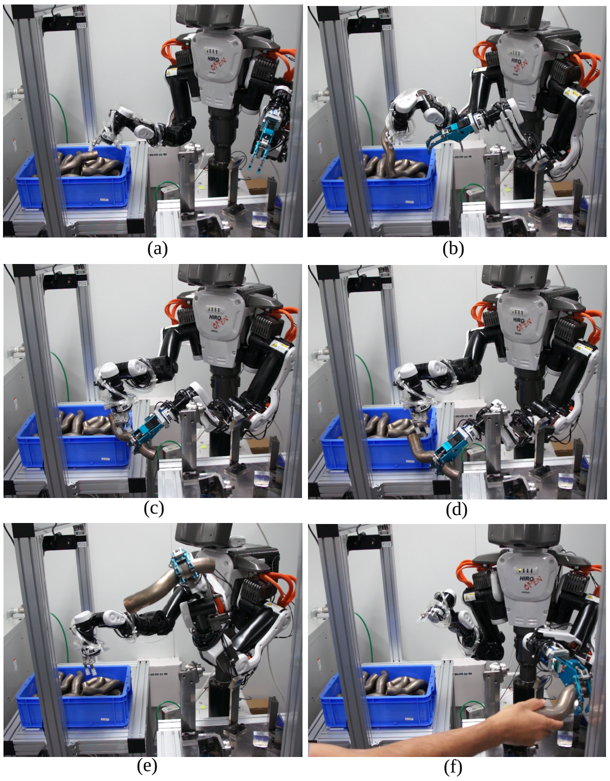

Fig. 4. An example of planned grasping motion