解 説

物体操作の行動計画技術

Object Reorientation and Manipulation Planning

万 偉 偉* *産業技術総合研究所

Weiwei Wan* *National Inst. of AIST

1. Introduction

Using robots to manipulate objects is essential to next-generation manufacturing. In the past decades, industrial robots were used in welding, painting, and pick-and-place in structured environments. These applications were successful but involved little manipulation. The problems related to them were mainly from control and had been intensively studied. In contrast, manipulating objects is more about AI (Artificial Intelligence) planning and is unsolved. Factories are still employing human beings to do object manipulation. For example, the robotic assembly pipeline in Glory won world-wide attention in 2013, however human workers have to prepare well posed and equally arranged objects beforehand; Toyota is one of the world-leading automobile giant and is employing thousands of robots in manufacturing, however its car assembly line involves irreducible human workers to pick out parts from boxes and perform assembly; The aeroplane giants, both Boeing and Airbus, are employing human workers to reorient components and install them; The rethink robotics company is assembling baxter robots by employing human beings, instead of using their robot products, etc. Therefore, solving object manipulation problems is essential to further promote industry robots.

This paper studies object manipulation and presents a planning framework for robotic manipulation planning. The framework is divided into three levels where in the highest level, the planning is symbolic deduction. In the middle level, the planning is object orientation. In the lowest level, the planning is motion planning. The paper focuses on the middle level (ob-

Fig. 1 Some perspective applications of our framework

ject orientation, Section 2 and 3) and explores the whole framework by expanding it to high-level and low-level planners (Section 4).

The framework is expected (but not limited to) to solve the object manipulation problems shown in **Fig. 1**, which includes (1) packing objects, (2) assembly, and (3) using tools. The common point of these problems is to find a sequence of robot motion that can reorient an object from an initial pose to a goal pose. The paper studies the object reorientation and manipulation planning problem. It also discusses different system scales and summarizes the knowledge we learnt during implementing the framework over the years at AIST (Section 5) to bring some insights to practical systems.

2. Object Reorientation

We formulate the problem of object reorientation as follows. **Input:** 1. The initial and goal pose of an object. 2. The kinematic model of a robot manipulator. 3. A position to do regrasp. **Output:** A sequence of grasp configurations and robot poses to reorient the object from the initial pose to the goal pose.

An example is shown in **Fig. 2** where the goal is to reorient a T-shape block. The initial and goal poses of the block are marked as “start” and “goal” in input part of the figure. The robot and its hands are Kawada Nextage and Robotiq 85. The position to do regrasp is marked as a red circle. The output part shows the planned grasp configurations and robot poses to reorient the object. Regrasp is the main technique used in object reorientation. Readers are recommended to see [1]~[8] to get a chronological view of the develop-

原稿受付 2016 年 9 月 30 日

キーワード: Grasping, Manipulation, Object Reorientation
*〒 305-8560 つくば市梅園 1-1-1

*Tsukuba-shi, Ibaraki

Fig. 2 Definition of the object reorientation problem

Fig. 3 (a.1-4) Grasp planning (a.1) Cluster facets (a.2) Find parallel facets and sample the parallel facets to generate contact points (a.3) Pose the finger pads to the sampled contact points and sample around rotational axis (a.4) Remove the collided grasps (b.1-3) Placement planning (c.1-2) Using the placements and grasps to build a regrasp graph

ment in regrasp planning.

The method used to solve the problem includes three components, namely placement planner, grasp planner, and regrasp graph solver :

1) Grasp planner : Grasp planner computes the accessible grasps of an object. The grasp planner uses a sample-based approach (Fig. 3 (a.1-5)). For each facet of the object model, the grasp planner sample contact points and check if the correspondent friction cones form force closure and the finger pads could be posed to the contact points without collision.

2) Placement planner : Placement planner computes the stable placement of an object on a table. For example, the tube has four stable placements which are shown in Fig.3 (b.2). Each stable placement is essentially a pose of the object. At these poses, the projection of the com (center of mass) of the object on the table is always inside the convex hull of the contact points. The accessible grasps of the object computed by grasp planner is trimmed to ensure hands do not collide with the table surface. The trimmed grasps are marked in light color in Fig.3 (b.3) (The dark part is the collided grasps). The trimmed grasps are the accessible grasps

Fig. 4 Search the regrasp graph to find a sequence of grasp configurations and robot poses that reorient the object from start to goal

of placements.

3) Regrasp graph solver : Regrasp graph solver involves two steps. In the first step, it builds a regrasp graph using the stable placements and their accessible grasps, the initial and goal poses of the object, and the position to do regrasp (see Fig. 3 (c.1-2)). In the second step, the regrasp graph solver searches the regrasp graph to find a sequence of stable poses and some accessible grasps to pick-up and place-down the object from these poses (see Fig. 4).

3. System Scale

Section 2 used a table to do regrasp. It is not efficient since regrasp is constrained by the limited number of stable poses on a table. The method may fail to find a path connecting the start and the goal on the regrasp graph. It may also return a long path where a robot have to do regrasp many times, increasing the chance of failure during execution.

The solution to deal with the inefficiency is to change the system scale. Here are some possibilities: (1) Regrasp using a tilting plane. (2) Regrasp using a support pin. (3) Regrasp using a hand fixed to the environment. (4) Handover using dual-arm or multi-arm robots.

These different scales increase the number of stable poses and hence increase the connectivity of the regrasp graph. They improve the efficiency of object reorientation by shortening the path found by the regrasp solver.

To implement these different scales, the placement planner must be replaced by a state planner: The feasible states on a tilting plane must be computed as the stable poses of an object lying on the tilted surface; The feasible states on a pin must be computed as the stable poses of an object leaning against the pin tip; The feasible states on a hand fixed to the environment must be computed as poses with accessible grasps to that hand; The feasible states of dual-arm handover must

Fig. 5 Different scales of object reorientation systems: (a) Regrasp using a tilting plane; (b) Regrasp using a support pin; (c) Regrasp using a hand fixed to the environment

Fig. 6 Using dual-arm handover to reorient objects

computed as poses with accessible grasps to both hands of the two arms.

Fig. 5 shows the different scales of systems we implemented using a simulator for the ABB IRB140 industry robot. We studied their efficiency and found regrasp using a support pin is not worse than using a hand fixed to the environment. **Fig. 6** shows the dual-arm handover implementation using a Nextage robot. We compared regrasp using a horizontal surface and regrasp using dual arms, and found dual-arm handover is not necessarily better. Finding collision-free grasps for the both two hands is sometimes difficult.

4. Object reorientation in the scope of manipulation planning

The object reorientation is one component of a whole manipulation planning system. Its position is like a mid-level component shown in the dash frame of **Fig. 7**.

In the highest level, the planning is symbolic deduction using various reasoning techniques. An example is assembly sequence planning [9]~[11]. Assembly sequence planning finds a sequence of object poses which will be used as the starts and goals of object orientation planning. In the middle level, the planning is object orientation. In the lowest level, the planning

Fig. 7 A three-level manipulation planning system. The object orientation planning discussed in sections 2 and 3 is located in the middle. It receives input from high-level symbolic planners and provides data to low-level motion planners

is motion planning, which could be either probabilistic algorithms like PRM and RRT and optimization algorithms like CHOMP. The output of object reorientation (grasp configurations and robot poses) will be used as the starting and goal configurations of motion planners.

5. Making it practical

The manipulation research group at AIST has implemented several object reorientation and manipulation planning systems using C++ (Choreonoid plugin), Python, and Matlab. During the implementation, we learnt lots of experience in making the system practical. Some suggestions are as follows.

1) Soft gripper vs. Rigid gripper: Soft gripper is recommended to do object reorientation. One problem we encountered using rigid grippers is that the uncertainty from visual detection may lead to collision. Soft grippers are compliant and avoids the problems caused by collision. Note that the soft grippers must be able to exert large force and hold objects firmly. The object must have a deterministic pose when being grasped.

2) Deterministic motion planning vs. Probabilistic motion planning: Deterministic motion planning is recommended in the low level to find a motion between two grasp configurations and robot poses. It is difficult to take the signal cables into account using probabilistic motion planning algorithms (A random pose leads to random cable twists). We encountered unexpected failures where the motion planner found a path

but the signal cables knocked down the object during regrasp. In contrast, deterministic motion planners orchestrate the output of the object reorient planner by inserting predefined motion clips (motion primitives [12]), which greatly reduces the unexpected failures.

3) Grasp quality: During reorientation, the torque exerted by the mass of the object changes. We recommend assigning weights to accessible grasps using their ability to resist external torques so that the regrasp solver can choose the grasps that are robust to rotation (e.g. the object doesn't pivot during reorientation).

4) Placing down objects: Placing down objects to a stable pose requires all contact points between the object and the table are simultaneously reached. Misplacement results into wrong poses and impair the next grasp. We recommend improving the performance of placing down objects by using some tactile sensors covered on a table to detect the contacts between the object and the table. Note that tactile sensors on tables are preferable than torque sensors on the wrist.

4) Uncertainty: Some actions lead to large uncertainty while others lead to low uncertainty. For example, gripping an object from two orthogonal directions sequentially reduces the translational uncertainty on the plane defined by the two orthogonal directions. This is a partially observable Markov decision process (POMDP) and is also highly recommended [13].

6. Conclusions and further reading

This paper presented a planning framework to do quasi-static object manipulation. The framework is composed of three levels which take different roles in symbolic reasoning and geometric planning. Different implementations were developed to analyze system scales and demonstrate the efficacy of the framework. Systems could be made practical by considering the experience summarized in section 5. The paper is a collection of several previous work we did at AIST. Readers are recommended to refer to [14]~[16] to further study the technical details and system performance of our past implementations.

References

- [1] P. Tournassound, T. Lozano-Perez and E. Mazer: "Regrasp-

- ing," Proc. ICRA, pp.1924–1928, 1987.
- [2] F. Rohrdanz and F. M. Wahl: "Generating and Evaluating Regrasp Operation," Proc. ICRA, pp.2013–2018, 1997.
- [3] K. Cho, M. Kim and J.B. Song: "Complete and Rapid Regrasp Planning with Look-up Table," J. Intell. Robot. Syst., vol.36, no.4, pp.371–387, 2003.
- [4] Z. Xue, J.M. Zoellner and R. Dillmann: "Planning Regrasp Operations For a Multifingered Robotic Hand," Proc. CASE, pp.778–783, 2008.
- [5] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner and R. Dillmann: "Humanoid Motion Planning for Dual-arm Manipulation and Regrasp Tasks," Proc. IROS, pp.2464–2470, 2009.
- [6] J.P. Saut, M. Gharbi, J. Cortes, D. Sidobre and T. Simeon: "Planning Pick-and-place Tasks with Two-hand Regrasping," Proc. IROS, pp.4528–4533, 2010.
- [7] S.J. Mousavi and E. Masehian: "A New Taxonomy of Robotic Regrasp Planning Approaches and Methods," Proc. ICROM, pp.90–95, 2013.
- [8] P. Lertkultanon and Q.C. Pham: "A Single-query Manipulation Planner," IEEE Robot. Autom. Lett., vol.1, no.1, pp.198–205, 2016.
- [9] L.S.H. de Mello and A.C. Sanderson: "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences," IEEE Trans. Robot. Autom., vol.7, no.2, pp.56–61, 1991.
- [10] R.H. Wilson and J.C. Latombe: "Geometric Reasoning About Mechanical Assembly," Artif. Intell., vol.71, no.2, pp.371–396, 1994.
- [11] R.A. Knepper, T. Layton, J. Romanishin and D. Rus: "IkeaBot: An Autonomous Multi-Robot Coordinated Furniture Assembly System," Proc. ICRA, pp.855–862, 2013.
- [12] K.K. Hauser, T. Bretl, K. Harada and J.-C. Latombe: "Using Motion Primitives in Probabilistic Sample-Based Planning for Humanoid Robots," Proc. RSS, 2006.
- [13] L.P. Kaelbling and T. Lozano-Perez: "Integrated Task and Motion Planning in Belief Space," Int. J. Robots. Res., vol.32, no.9–10, pp.1194–1227, 2013.
- [14] W. Wan and K. Harada: "Developing and Comparing Single-arm and Dual-arm Regrasp," IEEE Robot. Autom. Lett., vol.1, no.1, pp.243–250, 2016.
- [15] C. Cao, W. Wan, J. Pan and K. Harada: "Analyzing the Utility of a Support Pin in Sequential Robotic Manipulation," Proc. ICRA, pp.5499–5504, 2016.
- [16] W. Wan and K. Harada: "Achieving High Success Rate in Dual-arm Handover Using Large Number of Candidate Grasps," Adv. Robotics., vol.30, no.17–18, pp.1111–1125, 2016.

万 偉偉 (Weiwei Wan)

Weiwei Wan is a researcher of National Institute of Advanced Industrial Science and Technology (AIST). He received M.Sc.Eng. from Peking University in 2010 and Ph.D. from the University of Tokyo in 2013. He worked as a JSPS postdoc and did his visiting research at Carnegie Mellon University from 2013 to 2015. His research interest is grasping and manipulation.