Master Thesis

Nicola Covallero

April 10, 2016

State of the Art Manipulation Planning

cite David's works

still to add reference to the ones in the pushing.odt file

Many manipulation planning approaches (see [18] for an overview) assume that the task can be treated as a geometric problem, with the goal to place the objects in their desired positions. Planning is essentially done with a mixture of symbolic and geometric states, this requires a set of symbolic predicates that correspond to geometric relationships. In particular to the best of my knowledge, all the planners for cluttered scene are based on a mixture of symbolic and geometric predicates, or based only on geometric ones.

Dogar and Srinivasa [9] proposed a framework for planning in clustter scenes using a library of actions inspired by human strategies. They designed a new planner that decides which objects to move, the order to move them, where to move them, and it chooses the manipulation actions to use on these objects, and accounts for the uncertainty in the environment all through this process. The planner first attempts to grasp the goal object, than if it is not possible it identifies what is the object that obstacles the action, then such an object is added to a list of the objects that have to be moved. They moved an object in whatever position such that makes the goal feasible. Despite this, such a planning strategy can be used only in the design of a new planner, which is something we want to avoid.

To grasp they used the Push-grasping action [8] , which is a robust way of grasping objects under uncertainty. It is a straight motion of the hand parallel to the pushing surface along a certain direction, followed by closing the fingers. For the pushing directions they used a resolution of $\pi/18$ rad (i.e. 36 different directions) and use a predefined set of 9 hand aperture values. However their cluttered scenes is intended to be a scene with separated and well known objects.

In [12] the authors proposed to apply a linear temporal logic (LTL) planner to a manipulation planning task in cluttered scenes, but it suffers from poor runtime and the LTL specification is complex. Moreover to the scope of this work the temporal specification is not needed since it is indirectly encoded in the clutter scene composition.

An interesting planner which mixes symbolic and geometric predicates is aSyMov [3]. It is probably the most well known planner which a task to move objects considering how their location will then affect the possible robot's paths. To do that they combine a symbolic planner with a probabilistic roadmap [26] for geometric planning.

A recent alternative proposed by Mösenlechner and Beetz [22] is to specify goals symbolically but evaluate the plan geometrically. The idea is to use a high-fidelity physics simulation to predict the effects of actions and a hand-built mapping from geometric to symbolic states. Planning is conducted by a forward search, and the effects of actions determined by simulating them, and then using the mapping to update the symbolic state. This could be look promising to the scope of this thesis but still implies the design of a new planner.

In [19], Lonzano-Peréz et al. They use a simple task-level planner, in which operators are described with two types of preconditions: symbolic and geometric. They proposed a strategy for integrated task and motion planning based on performing a symbolic search for a sequence of high-level operations, such as pick, move and place, while postponing geometric decisions, based on the

CSP (Constraint Satisfaction Problem) technique. **Their technique allow working directly with state of the art planners.**

In [4] the authors address a problem similar to the one of this thesis. Pushing presents itself as a more intricate problem than grasping, a for grasping, after the object has been grabbed, it becomes a matter of solving a geometric problem. However, pushing an object carries uncertainty to the resulting state of the object. It is, nonetheless, important to be able to incorporate these two actions in the planning. For pushing they used a similar strategy to [21]. The planning algorithm also incorporates grasping, which, together with push actions, can perform a vaster array of tasks. It uses the concept of **reachability** [27] for each kind of action to quickly exclude impossible poses of the gripper of the planning stage, creating a reliable plan suitable for real-time operation. The authors model the planning problem through MDP (Markov Decision Process), discretezing the world in grid cells and assigning to each one a push and grasp vector definied by the reachability concept. Moreover each object is also classified through a simple primitive shape in order to have a proper map of pushing and grasping for each kind of considered object.

The majority of the state of the art for task planning in cluttered scene is focused on designing a new planner, why in this thesis state of the art planners ready to use want to be used.

Learning in Task Planning Machine learning techniques also have been applied to task planning. To plan complex manipulation tasks, robots need to reason on a high level. Symbolic planning, however, requires knowledge about the preconditions and effects of the individual actions. In [2] the authors proposed a practical approach to learn manipulation skills, including preconditions and effects, based on teacher demonstrations. It is difficult to solve most real world manipulation tasks by reasoning purely in terms of low-level motions due to the high-dimensionality of the problems. Instead, robots should reason on a symbolic level and appropriately chain the learned actions to solve new tasks. Such a planning step, however, requires knowledge of the important preconditions and effects of the actions. With few demonstration the authors proposed a method to teach the robot the precondition and effects of individual actions. Their method furthermore enables the robot to combine the learned actions by means of planning to solve new tasks that are more complicated than the learned individual actions. They focused their work on a class of manipulation actions where the preconditions and goals depend on spatial or geometrical constraints between the involved objects. They describe a scene as a collection of features and during the action demonstration the algorithm looks for the features that have a small change to identify the ones that can be predicates or effects of the considered action. They use an heuristic metric to define the variation of the features. The problem of this approach is that it moves the problem of identifying the right predicates to describe the problem in identifying right features that let a correct learning of the predicates. **We could consider this as further work**.

In [7] [6] the authors propose an approach for planning robotic manipulation tasks which uses a learned mapping between geometric states and logical predicates. The planner first plans symbolically, then applies the mapping to generate geometric positions that are then sent to a path planner. They try to fit a probability distribution function to the geometric states to map them to a symbolic state. This work sounds promising but the cluttered scene this thesis is going to treat is quite complex to be treated with such a method.

Interpreting the scene Focusing on symbolic planning, the research group of Artificial Intelligence and Robotics Laboratory of Istanbul Technical University, published some interesting researches suitable to the aim of this thesis. In [10] [23] [11] the authors propose a system which combines 3D recognition and segmentation results to create and maintain a consistent world model involving attributes of the objects and spatial relations among them. Unknown objects are modelled by using the segmentation output to determine their sizes and considering similarities with existing models to determine their shapes and colors. Then, these models are also stored as templates to be used for recognition along with the extracted attributes. They focused on the extraction of size, shape and color attributes as well as the following unary and binary spatial relations: on, on ground/on table, clear and near for object manipulation scenarios. These predicates are generated and updated over time to maintain a consistent world model. **For the aim of this thesis this paper is of particular interest because let to compute nicely some useful symbolic predicates** The *on* relation for a pair of objects is determined by checking whether their projections on the table plane overlap. For each pair of objects the *near* relation is determined by comparing the distance between the centers of these objects in each dimension with the sum of the corresponding sizes in that dimension.

Check fro backtracking techniques in planning, way to evaluate a plan

1 Segmentation

[16] interact with pile of objects with the aim to complement the segmentation algorithm for table clearance tasks.

2 Review Pushing

Review of pushing techniques

[13] object singulation

[14] object interactive perception and the same author did a work similar [15]

[20]

3 Review Grasping

Review of grasping techniques

In [28] they presented a uniform deterministic sequence and sets of sample over 2-sphere and SO(3).

In [5] paper of grasping

ROS object_manipulator

4 Usefull things

Temporal filtering to reduce the noise of the kinect [23].

5 Usefull Concepts

Closed world assumption [25] can be defined as having complete knowledge about the world, that is, the numbers and the attributes of all objects are known apriori.

6 What we have

- A naive pushing approach able to understand which object blocks the pushing action of another one. For the pushing action we could try to create a classifier in order to understand what it is the most likely pushing direction using geometric features. In this case we can evaluate different directions and wwith the feasible ones we could choose the best one. To push we could use or ProMP or DMP. When we have more possible pushing directions choose the one the moves the nearer the center of the robot's workspace. **Can we learn the pushing action? With a learning algorithm, not a classifier. Maybe we could create a classifier based on learning procedure.**

- Understand when an object is on top of another one

- Grasping? we miss how to decide if an object is graspable and if it is possible and, when it is not, understand if it is fault of an adjacent object. Strategy to resolve that:

  – Haf algorithm (very expensive) first on the object considering adjacent objects and then only on the object, so we can compare and understand if that was feasible.

  – HAF and then checking collision with the environment modelling in a simpler way the gripper, and detecting what are the objects that obstacle it

    With haf we also should test different rotations, this is too much expensive

  – AGILE: (not too expensive, but still more or less 2-3 seconds for whatever point cloud) and it selects just randomly the grasping poses. It is not a good choice doing that except using a very large number of sample, in that case we also have to use a point cloud reconstruction to resolve the problems seen in my work.

  – Naive approach: considering grasping it in the principal direction, it is computationally very cheap, and detecting for collision.

  – For objects that are not sorrounded by any other objects we should use a more powerfull algorithm in order to assure a good grasping pose detection.

The most of state of the art dealing with problems similar to this one manage the grasping action as unfeasible for the part of object that obstacle the arm, not the gripper, since they are usually dealing with uncluttered scenarios, where the objects are all separated among themselves. The idea here is to relax the grasping checking in order to speed up the translation process into symbolic level. Then to perform real grasping a reliable algorithm is used in order to get a good grasping, and checked for collision. If it is not possible the grasping we can deduce that the robot is not able to find grasp it, and take off from the goal such object and all the objects. Considering the case that such an object is on top of another one also the bottom object cannot be included in the goal, so it should be not included into the planner's goal.

7 Planners

8 Materials

8.1 Collision Detection Libraries

There exists several collision detection libraries, usally done for videogames such as: OZCollide [17], Bullet (available also for ROS framework)[1], Flexible Collision Library [24]. OZCollide has a poor documentation, try bullet or FCL.

- Classical planners
- Hierachical Planners
- Probabilistic Planners

9 Problem Description

10 Planner Structure

10.1 PDDL

11 Segmentation

12 Predicates

12.1 Block Predicates

12.2 On Top Predicates

12.3 Block Grasp Predicates

13 Pushing Action

14 Grasping Action

15 Real Robot

[18]

References

[1] Bullet `http://wiki.ros.org/bullet`, 2015.

[2] Nichola Abdo, Henrik Kretzschmar, Luciano Spinello, and Cyrill Stachniss. Learning manipulation actions from a few demonstrations. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1268–1275. IEEE, 2013.

[3] Stéphane Cambon, Fabien Gravot, and Rachid Alami. A robot task planner that merges symbolic and geometric reasoning. In Ramon López de Mántaras and Lorenza Saitta, editors, *ECAI*, pages 895–899. IOS Press, 2004.

[4] Rui Coelho and Alexandre Bernardino. Planning push and grasp actions: Experiments on the icub robot.

[5] Hao Dang and Peter K. Allen. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pages 1311–1317. IEEE, October 2012.

[6] Richard Dearden and Chris Burbridge. An approach for efficient planning of robotic manipulation tasks. In *ICAPS*. Citeseer, 2013.

[7] Richard Dearden and Chris Burbridge. Manipulation planning using learned symbolic state abstractions. *Robotics and Autonomous Systems*, 62(3):355 – 365, 2014. Advances in Autonomous Robotics — Selected extended papers of the joint 2012 {TAROS} Conference and the {FIRA} RoboWorld Congress, Bristol, {UK}.

[8] Mehmet Dogar and Siddhartha Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, October 2010.

[9] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. In Nick Roy Hugh Durrant-Whyte and Pieter Abbeel, editors, *Robotics: Science and Systems VII*. MIT Press, July 2011.

[10] Mustafa Ersen, Melodi Deniz Ozturk, Mehmet Biberci, Sanem Sariel, and Hulya Yalcin. Scene interpretation for lifelong robot learning. In *Proceedings of the 9th international workshop on cognitive robotics (CogRob 2014) held in conjunction with ECAI-2014*, 2014.

[11] Mustafa Ersen, Sanem Sariel Talay, and Hulya Yalcin. Extracting spatial relations among objects for failure detection. In *KIK@ KI*, pages 13–20, 2013.

[12] Keliang He, Morteza Lahijanian, Lydia E. Kavraki, and Moshe Y. Vardi. Towards manipulation planning with temporal logic specifications. In *2015 IEEE Intl. Conf. Robotics and Automation (ICRA)*, pages 346–352, Seattle, WA, May 2015. IEEE.

[13] Tucker Hermans, James M. Rehg, and Aaron F. Bobick. Guided pushing for object singulation. In *IROS*, pages 4783–4790. IEEE, 2012.

[14] Dov Katz. *Interactive perception of articulated objects for autonomous manipulation*. University of Massachusetts Amherst, 2011.

[15] Dov Katz, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Anthony (Tony) Stentz . Clearing a pile of unknown objects using interactive perception. In *Proceedings of IEEE International Conference on Robotics and Automation*, March 2013.

[16] Dov Katz, Arun Venkatraman, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Autonomous Robots*, 37(4):369–382, 2014.

[17] Igor Kravtchenko. OZCollide `http://www.tsarevitch.org/ozcollide/`, 2015.

[18] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, May 2006.

[19] Tomás Lozano-Pérez and Leslie Pack Kaelbling. A constraint-based method for solving sequential manipulation planning problems. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3684–3691. IEEE, 2014.

[20] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[21] Tekin A Mericli, Manuela Veloso, and H Levent Akin. Achievable push-manipulation for complex passive mobile objects using past experience. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 71–78. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[22] Lorenz Mösenlechner and Michael Beetz. Using physics- and sensor-based simulation for high-fidelity temporal projection of realistic robot behavior. In *AIPS*, 2009.

[23] Melodi Ozturk, Mustafa Ersen, Melis Kapotoglu, Cagatay Koc, Sanem Sariel-Talay, and Hulya Yalcin. Scene interpretation for self-aware cognitive robots. 2014.

[24] Jia Pan, Sachin Chitta, and Dinesh Manocha. Fcl: A general purpose library for collision and proximity queries. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3859–3866. IEEE, 2012.

[25] R. Reiter. A logic for default reasoning. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 68–93. Kaufmann, Los Altos, CA, 1987.

[26] T. Siméon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Advanced Robotics*, 14(6):477–493, 2000.

[27] Nikolaus Vahrenkamp, Tamim Asfour, and Rudiger Dillmann. Robot placement based on reachability inversion. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1970–1975. IEEE, 2013.

[28] Anna Yershova and Steven M. LaValle. Deterministic sampling methods for spheres and SO(3). In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pages 3974–3980, 2004.