Generalizing Task Parameters Through Modularization

Renaud Detry   Martin Hjelm   Carl Henrik Ek   Danica Kragic

***Abstract*— We address the problem of generalizing manipulative actions across different tasks and objects. Our robotic agent acquires task-oriented skills from a teacher, and it abstracts skill parameters away from the specificity of the objects and tools used by the teacher. This process enables the transfer of skills to novel objects. Our method relies on the modularization of a task's representation. Through modularization, we associate each action parameter to a narrow visual modality, therefore facilitating transfers across different objects or tasks. We present a simple experiment where the robot transfers task parameters across three tasks and three objects.**

I. METHODOLOGY

This paper addresses the problem of planning task-oriented grasps in an open-ended environment [6], [10]. We aim to produce a robotic agent that is able to plan how it needs to grasp a tool or an object to the end of executing a given task. A naive solution to that problem consists in hardcoding grasps for every object-task combination. While such a solution may work in a restricted environment, it quickly becomes intractable as the number of objects and tools grows. Instead, we suggest to teach tasks to the robot with a limited number of objects, and provide the robot with means of generalizing its experiences to novel combinations of objects and tasks. Our approach relies on a modularization of a task's grasp model. We isolate different components of the model, and we let the robot learn the perceptual parameters of each component individually. This modularization allows us to link each component to a restricted *family* of visual features that are directly relevant to it. Families of features may correspond, for instance, to 3D surface-related features [2], [11], 2D photometric features [8], volume moments, etc. By linking each component to a restricted family of features, we make it easier to transfer them to new situations.

This paper provides a simple yet functional illustration of the idea discussed above. For the purpose of this illustration, we argue that the minimal requirements for task-oriented grasping are twofold. First, the robot needs to leave the *functional site* of the tool or object available for the task. If the robot needs to hand an object to another agent, it needs to leave a part of the object available for the other agent to grasp. If the task is to pour from a recipient, the robot should stay clear of the opening of the recipient to avoid contact

R. Detry is with the University of Liège, Belgium. Email: renaud.detry@ulg.ac.be. M. Hjelm, C. H. Ek and D. Kragic are with the Centre for Autonomous Systems and the Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {martinhjelm,chek,danik}@csc.kth.se.

This work was supported by the Swedish Foundation for Strategic Research, the Belgian National Fund for Scientific Research (FNRS), the Swedish Research Council, and the EU project TOMSY (IST-FP7-Collaborative Project-270436).

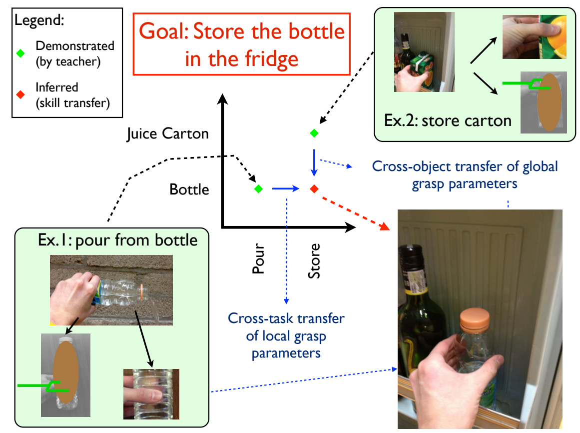

Fig. 1: Generalizing task parameters to plan a task with a novel object.

with the liquid. In the following, we refer to this aspect of planning as the *task constraints*. The second requirement for task-oriented grasping is to place the gripper onto a part of the object whose shape is such that the resulting grasp will firmly bind the object to the gripper. We refer to this aspect as the *gripper constraints*.

Following our definition of the minimal requirements for planning a task-oriented grasp, we devise a planner modularized into two components. The first component is responsible for encoding task constraints, while the second component encodes gripper constraints. Grasps are planned by maximizing the joint response of the two components. Before further formalizing these two components (next section), we illustrate the concept of our approach with an example shown in Fig. 1. This figure is organized around a simplified representation of the product space of tasks and objects. The robot is taught two task instances, namely pour from a bottle, and store a carton in a fridge door. The robot is then asked to store the bottle into the fridge door, a task/object combination that it has not been taught. However, from its experience with the carton, the robot learned that, in order to *store*, it needs to grasp the object near its top, to avoid colliding with the door while inserting the object. "Grasping near the top" is a task constraint that is potentially transferrable to other objects, including a bottle. Yet, the experience acquired with the carton will not allow the robot to adequately grasp the bottle, as the shape of the carton and the bottle differ substantially. A set of finger placements that are compatible with the cylindric shape of the bottle are instead derived from the first task instance that has been taught to the robot. From its experience with the bottle, the robot has learned how to place its fingers around a cylindric shape. The placement of the fingers around the cylindric shape of the bottle is not necessarily specific to the action of pouring liquid. The robot eventually combines parts of the experience gained through two different task instances to plan a grasp for a previously unobserved task/action combination. Transfers such as the one discussed here will naturally not always lead to a coherent plan. They will however reduce the search space that needs to be explored to adapt to a changing environment.

II. EXPERIMENT

The previous section introduced a grasp planner made of two components, one encoding task constraints, and the other one encoding gripper constraints.

Models encoding what is referred to here as gripper constraints have been studied on multiple occasions. A popular approach is to model the shape of graspable object parts, either from 3D data [2], [4], [5] or 2D images [9]. In the following, we encode grasp constraints using our previous work [2]. From a few demonstrations, the robot learns the shape and size of parts by which objects are often grasped, and how these parts should be grasped, yielding a dictionary of prototypical graspable parts. To plan a grasp on a new object, the robot matches each prototypical part to the partial point cloud of the new object, and selects the grasp of the best-matching part.

Models encoding task constraints have also been discussed in the literature. Niekum et al. [7] studied a method for segmenting tasks in a human-demonstration feed and generalizing an abstract representation from multiple demonstrations. Gienger et al. [3] introduced *task maps*, which encode object grasps that are suitable for a specific task. Dang et al. [1] presented a representation that allows the robot to encode tactile and kinematic parameters related to a task, and learn such representations in simulation. Ying et al. [12] addressed both task and gripper constraints. The authors modeled gripper constraints by representing the inner shape of the gripper, and task constraints by encoding the forces required for a task. The robot planned a grasp by finding an object part whose shape matches the gripper's, and by comparing the forces that can be exerted by the gripper, to the forces required for the task.

In this paper, we opted for a conceptually simpler approach, by which we encode task constraints by modeling the position of a grasp with respect to the main axis of the object. The main axis of an object is computed from the principal components of the (possibly partial) point cloud of the object. The axis is normalized to unit length, and the fraction of its height at which the grasp is applied forms our task constraint. Despite its simplicity, this model can represent key constrains of many different tasks.

When instructed to execute a task with a novel object, the robot searches for a grasp, i.e., for a wrist pose, that jointly maximizes the two constraints defined above. Grasp searching is done stochastically, by defining a cost function from the task constraint, and including it in the search

Fig. 2: Task-oriented grasping experiment. The first column shows instances of a pouring task. The second column shows shaking-oriented grasps. The third column shows grasps for storing the objects on a low shelf. Green instances are demonstrated to the robot. The robot then transfers parameters across tasks and objects to plan the instances shown in red.

procedure of our previous work [2]. In the experiment below, task-constraint costs are proportional to the distance of the wrist position to the fraction of the object's main axis encoded in the task constraint.

We provide a proof-of-concept experiment in which we consider three different tasks, and three different objects. The tasks are pouring, shaking, and storing. The three objects are those shown in Fig. 2. We demonstrated three task instances to the robot: pour from the mashed potatoes box, shake the salt cylinder, and store the soap bottle. Task constraints were computed as explained above. The robot learned that pouring needs to be done by grasping the object at the bottom end of its main axis, to avoid blocking the hole by which the content of the container flows. Shaking is done by grasping as close as possible to the center of gravity of the object. In order to store an object on a low shelf, the robot must grasp the object near its top. Gripper constraints could have been learned directly from these three examples. However, we instead imported the dictionary of prototypes created in our previous work [2].

Given the three training (green) instances of Fig. 2, the robot successfully inferred the other possible combinations of tasks and objects. For example, the task constraint of the top-left instance of Fig. 2, combined to the gripper constraints (in the form of the dictionary of prototypical parts), led to the planning of a pouring-oriented grasp on the salt cylinder, shown in the leftmost image of the second row.

In future work, we plan to incorporate more perceptual features into the task constraint model, and learn those features that are relevant for modeling different tasks. We also plan to embed this method in a larger framework with the aim of efficiently reducing the task/object search space in autonomous task discovery.

References

[1] H. Dang and P. K. Allen. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[2] R. Detry, C. H. Ek, M. Madry, and D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE International Conference on Robotics and Automation*, 2013. to appear.

[3] M. Gienger, M. Toussaint, and C. Goerick. Task maps in humanoid robot manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2758–2764, 2008.

[4] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE International Conference on Robotics and Automation*, 2012.

[5] O. Kroemer, E. Ugur, E. Oztop, and J. Peters. A kernel-based approach to direct action perception. In *IEEE International Conference on Robotics and Automation*, 2012.

[6] T. Lozano-Pérez, J. L. Jones, E. Mazer, and P. A. O'Donnell. Task-level planning of pick-and-place robot motions. *Computer*, 22(3):21–29, 1989.

[7] S. Niekum, S. Osentoski, G. Konidaris, and A. Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[8] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic Grasping of Novel Objects using Vision. *International Journal of Robotics Research*, 27(2):157, 2008.

[9] A. Saxena, L. Wong, and A. Ng. Learning grasp strategies with partial shape information. *Association for the Advancement of Artificial Intelligence*, 2008.

[10] H. Sugie, Y. Inagaki, S. Ono, H. Aisu, and T. Unemi. Placing objects with multiple mobile robots-mutual help using intention inference. In *IEEE International Conference on Robotics and Automation*, 1995.

[11] J. D. Sweeney and R. Grupen. A model of shared grasp affordances from demonstration. In *International Conference on Humanoid Robots*, 2007.

[12] L. Ying, J. L. Fu, and N. S. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):732–747, 2007.