The PR2 Workshop - Mobile Manipulation of Kitchen Containers

Jan Becker, Christian Bersch, Dejan Pangercic, Benjamin Pitzer, Thomas Rühr, Bharath Sankaran, Jürgen Sturm, Cyrill Stachniss, Michael Beetz, Wolfram Burgard

{pangercic, ruehr, sturmju, beetz}@cs.tum.edu

{Jan.Becker,Benjamin.Pitzer,Christian.Bersch}@us.bosch.com

bharath@seas.upenn.edu, {stachnis,burgard}@informatik.uni-freiburg.de

Abstract— **In this work we report about our efforts to equip service robots with the ability to robustly operate articulated containers such as refrigerators and drawers in kitchen environments. We identified three important aspects for such systems: (1) the ability to detect fixtures on an articulated object, (2) to robustly open and close them and (3) to store and retrieve information about these objects in the map. In particular, we detect grasping fixtures such as handles and knobs in 3D point clouds using a RANSAC-based plane detection and subsequent clustering approach. Further, we developed two types of controllers to operate articulated objects: the first controller is model-free and incrementally opens articulated models. The second controller is model-based and estimates both the kinematic structure and the kinematic parameters. Finally, we store the inferred articulation models into our knowledge processing system *KnowRob* readily available for later interactions. All software components have been tested on two PR2 (Personal Robot 2) robots located at TUM and Bosch and are freely available on `ros.org`.**

I. Introduction

Autonomous mobile manipulators have demonstrated impressive new capabilities recently [1], [2]. Researchers have enabled robots in their labs to make pancakes, play pool, fold clothes, and more. The reasons behind this rapid progress are manifold: the growing interest in mobile manipulation, standardized platforms, open source code, and system integration, along with improvements in perception, control, learning, and planning. In this paper we report about the integrated effort of three PR2 Beta Sites on mobile manipulation of kitchen containers which includes opening and closing the doors and fetching the objects out of containers (see video `http://youtu.be/G3ygwaLflYg`). The effort gains maximum leverage from the open source algorithms in ROS and the standardized robotic platform PR2. **The application using herein presented techniques will be at display at Bosch's booth at the IROS 2011 conference and is available for download from the *drink_serving* [1] stack.**

II. Detection of Furniture Fixtures

In the application pipeline we first proceed by detecting the container faces using RANSAC-based segmentation in 3D point clouds as put forth in [3]. In order to detect fixtures we first find point clusters that are withing the polygonal prism of the container faces using euclidean distance measure

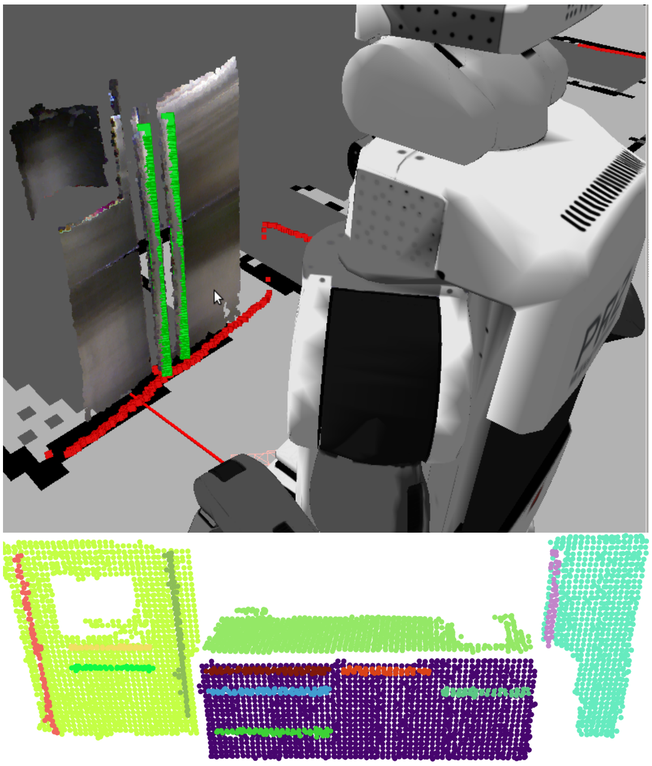

Fig. 1: Detection of handles at Bosch (top) and TUM (bottom).

and then fit RANSAC lines or circles to those clusters and thereby differentiate between handles and knobs. The geometric centroid is then used in order to grasp the fixture.

In order to open the door we developed two types of controllers that assume no a priori knowledge of the articulation model of the door.

III. Generalized Container Opening Controller

Firstly we present a general controller (see Algorithm 1 and code [2]) that makes use of the compliance of the PR2 robot's arms and the force sensitive finger tip sensors. Since the arms lack force sensors, the algorithm uses the Cartesian error of the end effector (commanded vs. actual position) to determine when the maximum opening is reached. The algorithm relies on the grippers maintaining a strong grasp

[1] `http://www.ros.org/wiki/drink_serving`

[2] `http://www.ros.org/wiki/ias_drawer_executive`

Algorithm 1: Controller for opening containers with unknown articulation model. Note: poses are stored as transformation matrices (translation vector and rotation).

Initialize p_0 = point on the handle candidate;
$p_1 = p_0 + n_{furnitureplane}; t = 0$
while *gripper_not_slipped_off AND Cartesian_error* $< threshold$ **do**
  if *d(p_{t+1},projection of robot footprint)* $< .1\ m$ **then**
    move_base(artif. workspace constr. for p_{t+1})
  move_tool(p_{t+1})
  stabilize_grasp() (see Figure 2)
  $Rel = p_0^{-1} * p_{curr}$ with current tool pose p_{curr}
  Extrapolate: Rel_s = scale $(Rel, (|Rel| + .05)/|Rel|)$
  $p_{t+2} = p_0 * Rel_s$
  t = t + 1
Return: Set of poses $P\{p_0...p_n\}$ representing the opening trajectory.

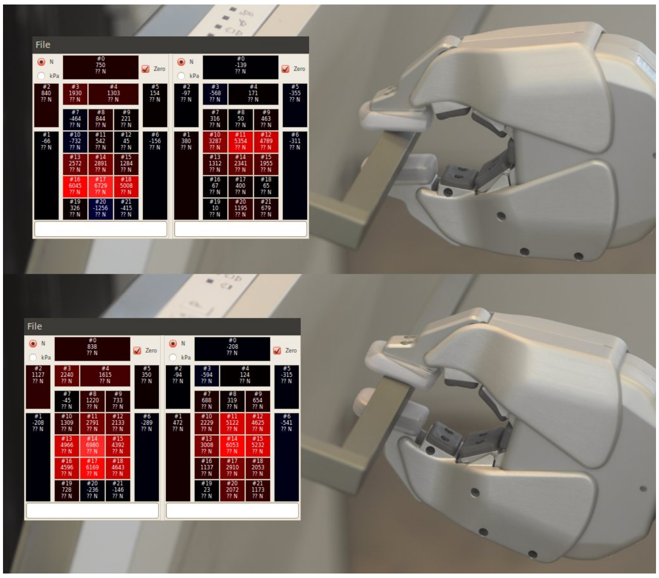

Fig. 2: The fingertip sensors are used to adjust the tool frame rotation to the rotated handle. Left part of the figure displays arrays of sensor cells on the PR2 robot's fingers. Asymmetry in the top-left part gives the measure of misalignment between the gripper and the handle.

while the arms are compliant. This way the mechanism that is to be opened steers the arm in its trajectory even when there is a considerable difference between the pulling and the opening direction. The robot also adjusts its base position if the door mechanism requires this. The controller records a set of poses with the stable (aligned) grasps and stores them in KnowRob [3] system for later re-use. The controller works reliably as long as the force required to open the container is lower than the limit the friction of the gripper tips imposes.

A particular problem when opening unknown containers is the possibility of collisions of a container with the robot.

[3] http://www.ros.org/wiki/knowrob

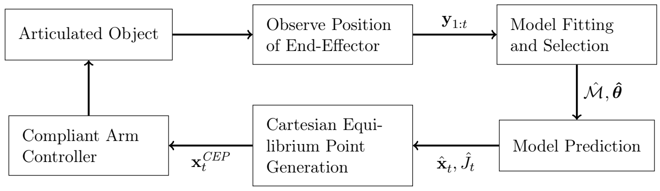

Fig. 3: Overall control structure. The robot iteratively estimates the kinematic model of the articulated object from the perceived trajectory of its end effector and evaluates it to generate the next Cartesian equilibrium point.

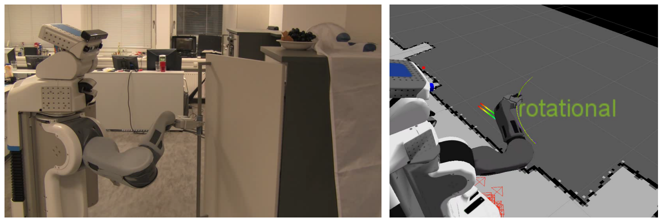

Fig. 4: The PR2 robot operates the fridge in the TUM kitchen and learns the kinematic model using Freiburg's articulation stack.

This could occur e.g. when a low drawer is being opened and pulled into the robot's base. Since the articulation model is not known, an a priori motion planning step is not possible. We thus propose the following heuristic: we exclude tool poses whose projections of the gripper to the floor fall close to or within the projection of the robot's footprint from the allowed workspace limit L of the gripper. This way, the robot tries to move backwards and prevents the collision.

IV. Kinematics-based Container Opening Controller

Second approach by Sturm et al. [4], [5] is depicted in Figure 3 which shows a block diagram of our controller for learning the kinematic model. We assume that the robot has already successfully grasped the handle of an articulated object and that a suitable initial pulling direction is known. The robot pulls in this direction using equilibrium point control (EPC) [6] and observes the resulting motion of its end effector. From this partial trajectory, it continuously (re-)estimates the kinematic model of the articulated object, that the robot uses in turn to predict the continuation of the trajectory. We deal with the workspace limits of the manipulator by a secondary controller that moves the omni-directional base of the robot so that the reachable volume of the manipulator is maximized. After the motion of the end effector has come to a rest, we estimate the range of valid configurations of the articulated object. In sum, this gives us the full kinematic model of the articulated object. Finally, we store the learned model in the *KnowRob* for later re-use as well. An example of this model learning step is visualized in Figure 4.

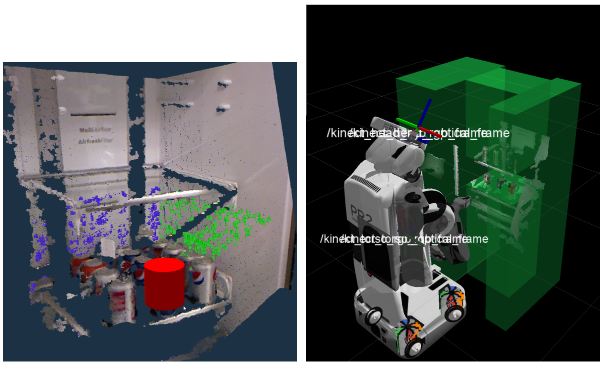

Fig. 5: **Left:** The red marker is used to represent the cluster extracted corresponding to the object of interest. The plane fits are illustrated by the colored overlay on the RGBD point cloud. **Right:** The collision model of the container is aligned with landmarks of the physical container.

V. Pose Invariant Object Detection in Containers

After the containers are opened, pose information relating to the objects of interest needs to be discerned to enable a successful grasp. Given the constrained manipulation space within these containers, the detection and pose information need to be accurate within a small margin of error.

To this effect, we use a priori knowledge of the internal structure of the container as a prior in our object detection pipeline. In the detection pipeline, once the internal structure of container is visible, planes are fit in the corresponding 3D point cloud. The intersection of perpendicular planes and their relative pose to the robot's base frame, gives us the relative pose of the container. This information, along with the knowledge of the planes is used as a prior for our object detection and clustering algorithms. We initially subtract the detected planes from the data to get the point cloud corresponding to the structure enclosed within the container walls. A SURF based nearest neighbor (FLANN [7]) template matching is performed to detect objects of interest in the scene around the location prior. To extract the pose and centroid of the cluster corresponding to the object of interest, the SURF matching results are used to augment the euclidean cluster segmentation of the point cloud (see Figure 5).

VI. Grasping Objects from inside Containers

In order to grasp detected objects from inside containers the robot has to avoid collisions with the container itself as well as with other objects while maneuvering its arm. For this, we use an RRT-based arm motion planner together with a collision map specifying obstacles for the robot to avoid. However, a perception-based collision map alone proved insufficient for computing collision-free arm motion trajectories as the robot cannot reliably detect a completely closed surface of the container, particularly when the container has reflective or transparent parts. Thus, we use a pre-specified collision model of the container, which is precisely aligned with the physical container object as visualized in Figure 5. This is achieved by locating the key landmarks such as the dominant planes of the container surfaces using 3D perception. Additionally, artificial collision obstacles are added to the collision map around the object to be grasped. This prevents the robot from colliding with objects that are hidden behind other objects and thus may have been missed during the object detection.

VII. Conclusions

The cooperation between different PR2 sites allowed us to join our efforts on building larger robotic applications. We combined our algorithms on door opening, perception of objects and planning, and extensively tested and validated our approaches at two different PR2 sites.

However, we also found that more generalization efforts are required so that these (existing) solutions become truly applicable in other environments. Therefore, we plan to direct our future research to generalize our approaches, for example, to deal with different shapes of handles, transparent doors and textureless objects. Authors would like to thank Willow Garage, Menlo Park, CA for providing and maintaining the PR2 robot and the ROS software. Without both of them the work presented above would not be possible.

References

[1] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA) 2011*, May 2011, pp. 1–8. [Online]. Available: http://graphics.berkeley.edu/papers/CusumanoTowner-BCD-2011-05/

[2] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the pr2," in *ICRA*, Shanghai, China, 05/2011 2011.

[3] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.

[4] J. Sturm, A. Jain, C. Stachniss, C. Kemp, and W. Burgard, "Operating articulated objects based on experience," Taipei, Taiwan, 2010.

[5] J. Sturm, C. Stachniss, and W. Burgard, "Learning kinematic models for articulated objects," *Journal on Artificial Intelligence Research (JAIR)*, vol. 41, 2011.

[6] A. Jain and C. Kemp, "Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control," Anchorage, AK, USA, 2010.

[7] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.