The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Contact-Reactive Grasping of Objects with Partial Shape Information

Kaijen Hsiao, Sachin Chitta, Matei Ciocarlie, and E. Gil Jones

Abstract— **Robotic grasping in unstructured environments requires the ability to select grasps for unknown objects and execute them while dealing with uncertainty due to sensor noise or calibration errors. In this work, we propose a simple but robust approach to grasp selection for unknown objects, and a reactive adjustment approach to deal with uncertainty in object location and shape. The grasp selection method uses 3D sensor data directly to determine a ranked set of grasps for objects in a scene, using heuristics based on both the overall shape of the object and its local features. The reactive grasping approach uses tactile feedback from fingertip sensors to execute a compliant robust grasp. We present experimental results to validate our approach by grasping a wide range of unknown objects. Our results show that reactive grasping can correct for a fair amount of uncertainty in the measured position or shape of the objects, and that our grasp selection approach is successful in grasping objects with a variety of shapes.**

I. Introduction and Related Work

As algorithms for autonomous operation are constantly evolving, complete robotic platforms with the ability to combine perception and action are starting to explore the rich set of applications available in unstructured environments. As part of this effort, this paper presents a set of algorithms that use real-time sensor data for grasping novel objects, with a focus on the two main components of an object acquisition pipeline: grasp selection and grasp execution.

The grasp selection (or grasp planning) task can be broadly defined as follows: given an object to be acquired using a robotic hand, find a combination of hand posture and position relative to the object that results in a grasp that is likely to resist expected perturbations (we note that the hand posture component is mainly applicable to dexterous hand designs). As grasps are, by nature, object-dependent, grasp selection in real-life applications is intrinsically tied to the sensory data available to the robot, especially if the robot is expected to operate on a variety of objects in multiple poses.

Grasp selection has been widely explored in recent work. Miller *et al.* [13] explored the use of shape primitives for object grasping. Their approach relied on having known models for the objects based on which grasps for the objects could be analyzed in the grasping simulation software *GraspIt!*. Srinivasa *et al.* [19] pre-computed grasps for objects and then executed them based on registration of the objects in the environment, and Geidenstam *et al.* [4] used box-based decompositions of simulated 3-D point clouds to learn 2-D grasping strategies for objects. The use of known 3-D models precludes these techniques from being used for grasp selection in unstructured environments. While there has also been recent work in constructing 3-D object models for grasping [11], [10], the quality of these models is still not good enough to be suitable for use in a grasping framework like *GraspIt!*. In contrast to the approaches that use 3D model information, our methods do not require a known model of the object to be grasped, and can be used even when sensor data is available only for parts of the objects.

Several systems have also used 2-D image data to try and compute grasps for unknown objects. Saxena *et al.* [18] attempted to learn grasp points directly from 2-D image data. Kamon *et al.* [9] also attempted to learn grasps from visual information. Pelossof *et al.* [15] used a learning based approach to estimate the quality of a grasp. However, no effort was made in these approaches to integrate reactive grasping to account for uncertainty in the location of the grasp. In [7], the work most similar to ours, a laser rangefinder was used to separate objects out from the background and pick them up. This approach allowed an assistive robot to pick up a wide range of everyday objects.

While range and imaging sensors may provide sufficient information for grasp selection, these sensors have shortcomings in the context of grasp execution. Grasp selection relies on (often static) information about the target object, which is well suited for sensors such as monocular or stereo cameras or laser range finders. However, some amount of position and/or modeling error is inevitable when using these technologies. Errors in grasp execution can also arise from imperfect calibration, environmental occlusion, or even imperfect control of the robot. A system that relies exclusively on ranged data during grasp execution is likely to encounter frequent problems.

Greater robustness in grasp execution can be achieved by incorporating and reacting to data acquired through non-ranged sensor modalities. Tactile sensors are natively well equipped for providing information during grasp execution, as they enable direct sensing of aspects such as contact force or relative velocity at contact points, without being affected by the occlusion problems inherent for range sensors. For example, Petrovskaya *et al.* [16] and Hsiao *et al.* [5] used tactile feedback to localize the position of a known object model for grasping. Platt *et al.* [8] and Felip and Morales [3] used tactile feedback to locally adjust a grasp, the first by minimizing the net force and torque of the grasp, and the second by trying to maximize the alignment and symmetry of the grasp. A haptic feedback approach to grasping with limited use of visual feedback was presented in [14]. Dollar *et al.* [2] used contact sensors on the insides of a compliant hand to center an object within the hand. Mayton *et al.* [12] used electric-field sensing to execute reactive grasps, which

Willow Garage Inc., Menlo Park, CA. Email: {hsiao, sachinc, matei, gjones}@willowgarage.com

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

Fig. 1. The PR2 grasping a bowl amongst obstacles.

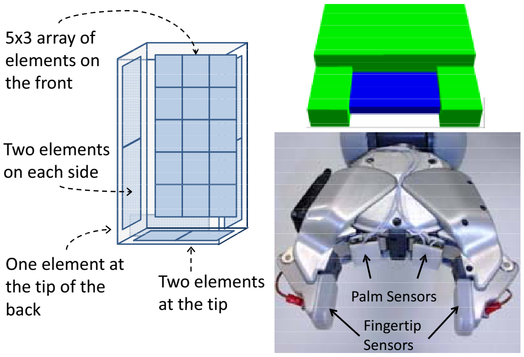
5x3 array of elements on the front
Two elements on each side
One element at the tip of the back
Two elements at the tip
Palm Sensors
Fingertip Sensors

Fig. 2. Left: diagram of the fingertip tactile arrays. Right: image of the PR2 gripper and its sensors, as well as a simplified gripper model used for grasp selection.

works well with objects whose di-electric constant is greater than the surroundings. Hsiao *et al.* [6] used information from optical sensors mounted on the fingertips to reactively conform to the shape of an object; this fails on shiny or transparent objects. Prats *et al.* [17] use visual, force and haptic feedback to design a controller that they demonstrated for door handle grasping and door opening. However, no attempt was made at online grasp selection.

In this study, we employ tactile-sensing-based algorithms to detect and react to contacts encountered during the execution of a grasp. By employing tactile sensing, we can sense contacts that indicate that a grasp will result in the object being pushed away, rather than enclosed, or that a grasp will be unstable and likely to fail when encountering disturbances. By sensing these error conditions and adapting the execution of the grasp, we increase the likelihood of achieving a successful, stable grasp. We note that such algorithms are largely independent of the grasp planning process, as many of the proposed planning methods will benefit from this type of "reactive" execution. However, integration with our grasp selection algorithm allows us to present and analyze a complete pipeline operating in a real environment. Furthermore, these two components complement each other well, as they both require minimal pre-computed knowledge of the world. As such, our pipeline is well suited for execution in unstructured settings, where we believe it will serve as a basis for more complex applications.

II. Hardware Platform

The hardware used for the experiments in this paper is the PR2 personal robot, a two-armed robot with an omnidirectional base. It has an extensive sensor suite useful for mobile manipulation, including a tilting laser scanner mounted to the head, two stereo cameras, an additional laser scanner mounted on the base, and a body-mounted IMU. One of the stereo cameras has narrow-angle lenses to generate accurate depth information for objects close to the robot, while the other has wide-angle lenses to generate depth information over a wider field of view. The narrow stereo sensor is used in this work to generate a 3-D representation of the objects to be grasped and their environment. Encoders on each joint also provide continuous joint angle information.

Figure 2 shows the parallel jaw gripper mounted on both arms of the robot. The gripper has a single actuator consisting of a brushless DC motor with a planetary gearbox and an encoder. Each gripper's fingertips are equipped with a capacitive sensor consisting of 22 individual cells. The 22 cells are divided between a $5{\times}3$ array on the parallel gripping surface itself, 2 sensor elements on the tips of the fingertips, 2 elements on each side of the fingertip, and one on the back. These capacitive sensors measure the normal pressure applied in each sensed region. An additional, custom set of contact sensors were mounted on the palm of the robot and were used to detect contact between the palm and an object.

III. Grasp selection

Our grasp selection algorithm operates using depth data obtained from a single frame of the narrow-field-of-view stereo camera on the PR2. In this type of partial object data, it is typically the case that the most desirable grasps of objects involve at least one finger contacting the object in occluded space. This means that we must use a significantly different approach for grasp selection than methods that use full 3-D meshes. However, even when considering only the partial scan, it is still possible to select viable grasps. For instance, Saxena *et al.* [18] learn weights on features associated with good grasps in 2-D images. Jain and Kemp [7] select overhead grasps using a small set of simple heuristics: if the object is small enough to fit within the gripper, the robot grasps the object by the centroid, with the gripper oriented perpendicular to the direction of maximum variance; otherwise, the algorithm uses high points on the objects and aligns the gripper axis in the direction of the object centroid.

Our approach to grasp selection is similar to previous work in that we also employ a small set of simple heuristics to select grasps. However, as we wish to deal with objects and poses of objects for which the methods of Jain and Kemp are insufficient, we expand the heuristics to search for an entire ranked list of grasps for each object, selecting the highest-ranked grasp that is both reachable and collision-free. We also draw from the results of Balasubramanian *et al.* [1], who have shown that humans tend to select grasps with wrist orientations that are orthogonal to the object's principal axis and its perpendiculars; such grasps tend to be more stable than those chosen using randomized planners that do not value orthogonality. Our search method will use heuristics to search for orthogonal grasps that approach from the top and the side, in addition to overhead grasps such as those used in [7], ranking the found grasps using a small set of simple feature weights.

A. Object point clouds and bounding box finding

Our sensor processing approach begins with data consisting of a point cloud of an entire scene, obtained from the stereo camera equipped with a texture projector. We first identify the parts of a point cloud (referred to as point *clusters*) that are likely to belong to a single object. To this end, we make two simplifying assumptions: the objects are sitting on a flat surface (such as a table), and the minimum distance between two objects in a scene is at least 3cm. We then identify the table as the dominant plane in the scene, and remove its corresponding points. The remaining points are then clustered together based on their projection on the table plane, using a variant of the mean shift algorithm; clusters that exceed a pre-set threshold of 1000 points are kept and smaller clusters are discarded. The resulting clusters consist of point clouds of the visible surfaces of the objects. Due to the nature of our sensor, bright, opaque and matte objects tend to produce denser clouds, with less missing data. Our main focus in this study is the utilization of clusters in grasp selection, rather than the clustering method itself; as such, we have not performed a quantitative analysis of our clustering methodology. For a qualitative analysis, all the grasp examples presented in this paper also include images of their respective point cluster.

The first processing step on every individual point cluster is to compute a local, object-centric coordinate system. Since we assume that objects are resting on a surface, the vertical direction (relative to gravity) is always given special treatment and assigned to the local z-axis. We use PCA to compute the axes of minimum and maximum variance in the horizontal plane, and assign them to the x- and y-axis respectively. We then find the bounding box of the point cluster along those axes and define its center as the origin of the object coordinate system.

B. Searching for good grasps

Our grasp selection algorithms is based on two main components. The first component uses a set of heuristics to **generate** two sets of possible grasps, one containing the grasps that are considered to have the highest chance of success, and the second containing fall-back options to be attempted only if no grasps from the first set are feasible. We refer to these as the "preferred" and "fall-back" sets. The second component of the selection algorithm is used to further **rank** each set in order of the perceived chance of successful execution. Both components use as input the object point cluster and its bounding box, as well as a model of the gripper, and the location of the table surface.

The gripper model that we use, shown in Fig. 2, consists of four boxes: one for each of the two fingertips, one for the palm, and one for the space between the fingerpads. The fingertip and palm boxes (shown in the figure in green) are used to estimate collisions, while the gripper space box (shown in blue) is used to check if a part of the object is likely to be enclosed in the grasp. We note that the gripper space box is only half as wide as the fingertip width, in order to reward contacts that are centered within the fingerpads. We define the gripper closing direction to be the axis between the two fingerpads.

The grasp generator component uses the set of heuristics presented below, with all axes referenced in the local object coordinate system described in Section III-A:

- *top grasps*: the object is approached along the z-axis (vertical direction). The gripper closing direction is then aligned with either the x- or y-axis. We note that grasping around the center of mass of an object is preferable, as it reduces the torque about the gripper axis due to the object's weight. However, since we only see part of the object, we use the center of the object's bounding box as a proxy for the true center of mass. For both gripper orientations (along x or y), one grasp is generated, centered above the bounding box's center. If the width of the box along the selected axis fits inside the gripper, the grasp is placed in the preferred set; else, it is placed in the fall-back set. Multiple grasps are then generated by moving the gripper along the third remaining axis, and placed in the fall-back set.
- *side grasps*: the object is approached in the horizontal plane, along the x- or y-axes, using the following rule: if the size of the bounding box along the x-axis fits inside the gripper, the x-axis is aligned with the gripper closing direction, while positive and negative y are used as approach directions. Multiple grasps are then generated by sampling along the z-axis. These grasps are all placed in the preferred set. The process is then repeated with the roles of x and y reversed.
- *high point grasps:* a set of points are chosen at random within 2 cm of the top (positive z direction) of the cluster bounding box. For each point, the approach direction is along the z-axis, with the gripper aligned with the line that connects the chosen point with the bounding box center. These grasps are particularly useful for bowls and other rotationally-symmetric containers with rims.

Our method for determining the exact grasp point is the same whichever heuristic is used for generation: the search along the approach direction starts with the gripper fingertips just outside the object bounding box, then proceeds inward towards the object. Along the way, we look for grasps with at least 50 points within the gripper space box, storing the one with the largest number of points and stopping when collision is detected between one of the gripper collision boxes and either the point cloud or its supporting table.

C. Ranking grasps

Once the sets of grasps have been generated, we compute a weighted sum of a set of features to produce a numerical quality measure for each grasp. The feature set has been chosen as follows:

- *point_count*: the number of points within the gripper model's space box.

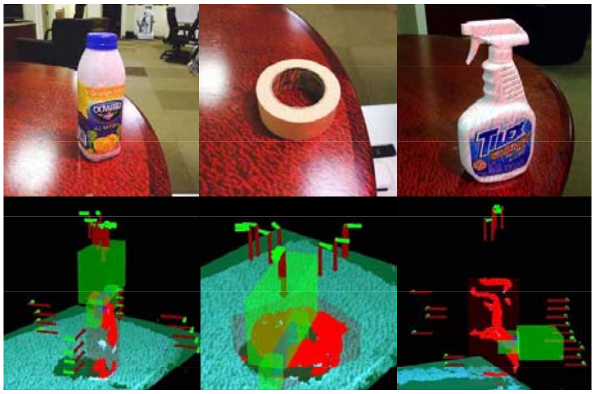

Fig. 3. Examples of grasps chosen by our selection algorithm. The gripper model is shown by the translucent green collision boxes and the blue space box, placed at the most desirable grasp. The red points show the object cluster, the box surrounding the object shows the object bounding box, and the arrows show other possible grasps, with the red arrow pointing in the approach direction and the green arrow along the gripper closing axis.

- *overhead* (binary): 1 for top and high point grasps, and 0 for side grasps. Favoring top grasps proves useful particularly when employing reactive grasping since unexpected contacts are more likely to be detected (and corrected) at execution time for overhead grasps. Side approaches can push light objects away while applying forces below the tactile sensors' sensitivity threshold.
- *centered* (binary): 1 for grasps that are not along the edge of the bounding box, 0 otherwise.
- *fits_in_hand* (binary): 1 for grasps where the bounding box of the object fits entirely within the gripper, for the chosen direction of the gripper closing axis, 0 otherwise.
- *side_dist*: the gripper's distance (in cm), along the axis orthogonal to the closing direction, from either the object center (for overhead grasps) or from 6 cm above the table (for side grasps). We note that in the case of side grasps, lower approach directions are preferred as they are less likely to to knock the object over.
- *palm_dist*: the distance between the fingertips and the object center along the approach direction.

Weights for the list of features (*point_count*, *overhead*, *centered*, *fits_in_hand*, *side_dist*, *palm_dist*) were empirically chosen to be (1, 500, 800, 1500, -100, -100). Each feature is multiplied by its respective weight and the result is summed, with higher-valued grasps preferred.

Several objects and the associated grasps that are found using our selection process are shown in Figure 3.

D. Grasp execution

Once the sets of preferred and fall-back grasps are ranked, individual grasps are analyzed based on their rank within the set. For each grasp, we use a pre-grasp position that is simply the grasp pose backed away by 10 cm along the approach direction. We then check whether there is a collision-free, kinematically feasible joint trajectory from the arm's current position to the pre-grasp, and also a consistent Cartesian trajectory to go from the pre-grasp to the grasp. If a grasp is deemed infeasible due to either kinematic constraints or arm collision with the environment, we move on to the next grasp in the list. Thus, having an entire list of possible grasps becomes more important as the scene becomes cluttered or as the object is placed farther away from the robot.

Once a grasp is chosen, the robot first moves the arm to the corresponding pre-grasp in an open-loop execution, then proceeds along the Cartesian path to the final grasp, and then finally closes the gripper. However, for this operation to result in a stable grasp when executed open-loop, there must be no errors in sensing, planning or trajectory execution. To cope with the possibility of such errors, we use reactive algorithms based on tactile sensor information. Using a pre-grasp pose ensures that the final stage of the grasp trajectory is always along the approach direction, which will allow us to reason about encountering unexpected contacts as described in the next section.

IV. Reactive grasping with tactile sensors

The reactive grasping component of our system uses the PR2's fingertip tactile arrays, palm contact sensors, and robot proprioception for a variety of simple, reactive behaviors. These behaviors are designed with several goals in mind: to recover from small positional errors, to grasp objects in a way that minimizes unwanted object pushing, and to locally adjust grasps that are likely to be marginal. Use of these behaviors is largely independent of grasp selection. Furthermore, if a planned grasp can be executed without adjustment it will, in nearly all cases, be indistinguishable from an open-loop execution. It is only if unexpected contact is made during execution or if the resulting fingertip contacts appear less stable than expected that we attempt to modify the robot's behavior. We will now discuss each reactive behavior in more detail.

A. Reactive approach

We employ Cartesian controllers to move the arm towards the grasp pose during execution. The controller gains are set to low values, to avoid applying substantial force to the object when making unexpected contacts.

During the approach our reactive algorithm acts as follows:

- if a contact is detected on the tip, side, or back of either fingertip, the gripper will back up and move sideways around the location of contact.
- after the initial contact, if the next contact is on the tip, side, or back of the opposite fingertip we assume that our sideways step was too large and move sideways in the opposite direction with a halved step size.
- if contact is detected only on the inner side of the fingertip, no reactive move is attempted; such contacts typically occur during approaches to objects that are nearly the dimension of the gripper opening.
- if contact is detected on either palm sensor, approach is halted; the assumption in these cases is that approaching further is impossible.

It is important to note that very light objects can be displaced without contacts registering on the tactile pads despite our use of low controller gains. We expect that improvements

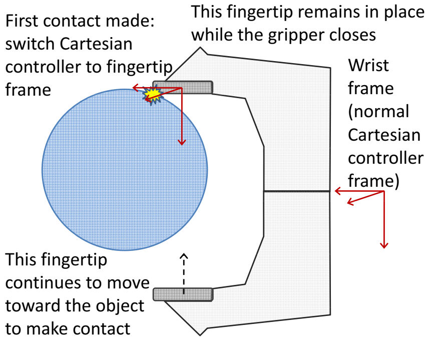
First contact made: switch Cartesian controller to fingertip frame
This fingertip remains in place while the gripper closes
Wrist frame (normal Cartesian controller frame)
This fingertip continues to move toward the object to make contact

Fig. 4. Compliantly closing around a cylinder.

in the sensitivity of tactile sensors will increase the efficacy of our reactive algorithms in coping with such objects.

B. Compliant closing

Once the gripper has reached the target grasp position and initiates closing actuation, there is the potential that one fingertip will hit the object before the opposite fingertip; in some this can result in a failed grasp, as the first fingertip may push the object out of the grasp before the second fingertip can secure the object. One method to prevent such failures is by keeping the first fingertip fixed in space while the second finger continues to close. We achieve this behavior by changing the target frame for the Cartesian arm controller, which is typically attached to the gripper's palm, to the frame of the contacting fingertip, as shown in Figure 4. This allows the second finger to compliantly move toward the object without the first finger applying force, as the Cartesian arm controllers compensate for the gripper closing.

C. Grasp adjustment

The next stage of our grasping approach is initiated after both fingertips have made contact with the object; in this stage the grasp may be adjusted to increase stability. In our approach we use the fingertip tactile arrays to determine whether the contacts are centered on the inner surface or are restricted to the edges of the fingertips; if the object is touching only the tips or edges of the fingertips, the grasp is likely to be unstable.

If we sense that the contact is isolated to the tips or edges we do the following:

- if contact is sensed only on the front part of either fingerpad near the tip, we attempt to adjust the grasp by opening the gripper and moving forward along the approach direction. If this motion results in contacts with the palm sensors, motion is halted.
- if we only sense contacts on the same edge of both fingertips, we attempt to adjust the grasp by opening the gripper and moving in the direction of the contacts.

Once the adjustments have resulted in centered contacts on both fingerpads, the grasping force is increased to the desired final grasp level. If increasing the force causes the contacts to become off-center, the gripper is opened and the adjustment behavior will be attempted again.

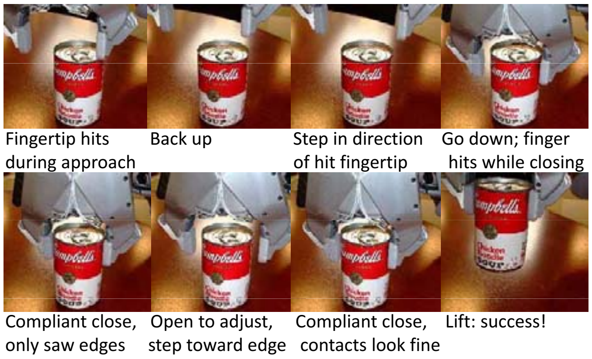
Fingertip hits during approach
Back up
Step in direction of hit fingertip
Go down; finger hits while closing
Compliant close, only saw edges
Open to adjust, step toward edge
Compliant close, contacts look fine
Lift: success!

Fig. 5. Example of a reactive grasp sequence using the complete set of reactive adjustments.

D. The full system for reactive grasping

We combine the behaviors for approach, compliant closing, and grasp adjustment into an integrated reactive algorithm for grasp execution described in Section III-D. If the grasp appears have been successful – the contact forces are centered on the fingerpads, the desired force level has been reached, and the gripper has a non-zero opening, which indicates the presence of an object – the object will be lifted off the table. If the gripper misses the object completely, a condition indicated by a fully-closed gripper with no palm contacts, the grasp target pose is moved further along the grasp approach direction, and the grasp is attempted again. We note that in some cases, successful grasps can be indicated by the presence of palm contacts and a fully closed gripper; this case could occur when an object's handle is completely enclosed between the fingers and the palm.

A reactive grasp therefore consists of a series of reactive adjustments, and the grasp itself can be tried multiple times. A detailed example of a grasp using the complete set of reactive adjustments is shown in Figure 5. Both the number of adjustments within a grasp, and the number of grasp attempts, are free parameters. In our experiments, we use two grasp attempts, each potentially employing four attempts at grasp adjustments. Our qualitative observations show that increasing the number of adjustments brings quickly diminishing returns, as failure is usually indicative of a poorly chosen grasp pose that can not be transformed into a stable grasp by local, reactive improvements.

E. Reactive Grasping Experiments

Our experimental analysis of the effectiveness of our tactile-based adjustment algorithm focused on assessing the range of positional errors that can be accomodated when using reactive versus open-loop grasping. The target object in our experiments was a cylindrical can; for a quantitative analysis, we executed a large number of overhead and side grasps. In all cases, the robot executed a grasp based on the expected position of the target. The true position of the target was then varied by up to 10cm in up both horizontal directions on the table.

Figure 6 shows the effect of reactive grasping for overhead grasps, as in the example in Figure 5. The chart shows

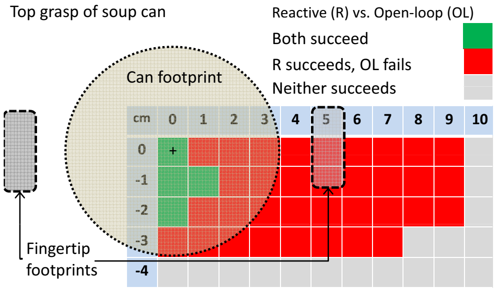

Fig. 6. Reactive vs. open-loop grasping for overhead grasps, under a wide range of positioning errors.

a top-view of the experimental scenario, where the circle represents the size and expected position of the target, and two rectangles are used to show the footprints of the fingertip pads at the nominal grasp position. The two axes of the chart represent displacement (in cm) of the true position of the target relative to the expected placement. Only displacements in one direction are shown for each dimension, since both the gripper and object are plane-symmetric with respect to both dimensions. The other quadrants can be inferred by reflecting the shown chart about the axes.

The green region is the area under which both open-loop and reactive grasping successfully grasp and lift the target object. The red region is the area under which reactive grasping continues to succeed, while open-loop grasping fails, either by dropping the object entirely or by performing a marginal, undesirable grasp of just the rim of the can. The grey region is the area under which neither algorithm succeeds. We note that even a small displacement of 1 cm can cause open-loop grasping to either fail or end up in a marginal grasp. On the other hand, the object can be displaced up to 9 cm in the direction of one fingertip, and reactive adjustment can still result in a stable grasp.

Figure 7 shows the results of comparing open-loop and reactive grasping for side grasps. Here we have only one plane of symmetry instead of two. The red and green regions are the same as before, and the dark grey region represents displacements that are large enough to cause collision with the pre-grasp position. Open-loop grasp succeeds for displacements up to 3 cm in the y direction, but fails for even a small displacement in the negative x direction as the object is squeezed out of the gripper. In contrast, the reactive grasping approach succeeds for displacements of up to 6 cm in the y direction and 8 cm at the farthest in the x direction. We note that adding more re-grasp attempts (beyond our standard limit of two) can increase the successful range in the x direction to as far as the arm is able to reach, as the robot will detect the absence of the object inside the gripper and try again further along the approach direction.

V. Experimental Results

Testing separate components of a grasping pipeline can help isolate the benefits of individual adjustments and algorithms. However, the most relevant analysis of robot behavior is obtained by quantifying its overall performance in unstructured settings. We thus evaluated the complete grasping pipeline, comprising both the grasp selector and the set of reactive behaviors, over a large set ($n = 30$) of common objects in a typical human setting.

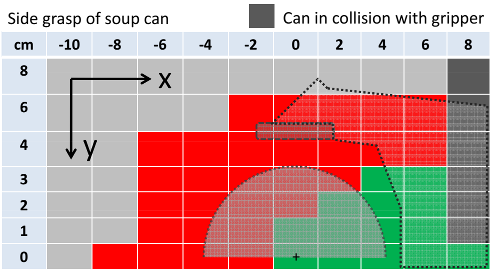

Fig. 7. Reactive vs. open-loop grasping for side grasps, under a wide range of positioning errors.

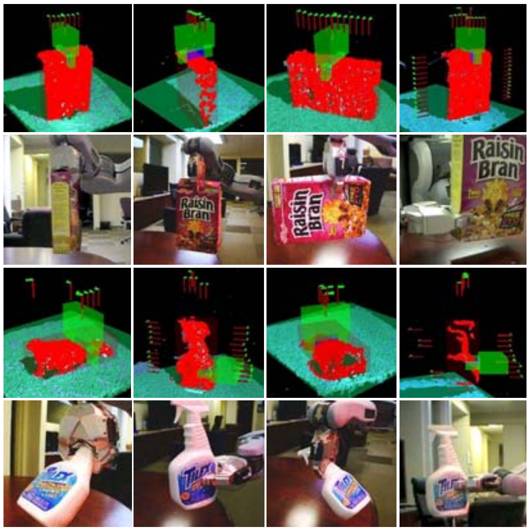

Fig. 8. Pose experiments: grasping a box of Raisin Bran and a Tilex bottle. Here we show four of the six different poses for each object.

Each object in the set was placed on a table in front of the robot. The pose of the object was chosen randomly (constrained only to lie inside the camera field of view); for each object, two different poses were tested. For each pose, the robot attempted to lift the object off the table using both reactive and open-loop grasping. In each case, the grasp was computed by the grasp selection algorithm presented in Section III-D. A set of experiments over the entire object set, showing point clouds, chosen grasps, and executed grasps for one of the object poses is shown in Figure 9. For two objects in this set (a tall, relatively wide Raisin Bran cereal box and an irregularly shaped Tilex liquid dispenser) we also performed grasps at four additional poses; results are shown in Figure 8. The complete set of experiments thus comprised 68 object/pose combinations.

Reactive grasping succeeded in stably grasping and lifting the object in 66 out of 68 attempts. Open-loop grasping was successfully executed 60 out of 68 times. Overall, the combination of grasp selection and reactive adjustment showed a high reliability level over a wide range of target objects and poses, using only run-time sensor data and no pre-computed knowledge base.

Additional insights can be obtained by analyzing the failure cases for both grasp execution modes. In the case of reactive grasping, one failure was recorded on a transparent red glass, which was nearly invisible to the stereo camera. As a result, the arm collided with an unseen part of the object while moving towards the pre-grasp pose. The only other failure case was a cup, for which a grasp was chosen that ended with the gripper fully closed around the cup handle. While such a grasp can be desirable, the lack of fingerpad or palm contacts led to an attempted adjustment, which in turn resulted in a failed grasp.

The non-reactive grasping method failed for both poses of the Pantene bottle, and on one pose for each of the following objects: two transparent glasses, soap bar, tea box, cereal box, and Tilex bottle. These objects show that reactive grasping is most useful in the case of objects, such as boxes or jars, that approach the maximum aperture of the gripper in width. This is true even when using a calibrated system, such as the PR2 robot, where the difference between stereo camera readings and joint proprioception is within one or two centimeters in the main workspace of the arms.

In a final set of experiments, the grasp selector for unknown objects with reactive grasp adjustment was integrated into a more complete grasp pipeline. GraspIt! was used to select grasps for objects recognized (using an ICP-like algorithm on the segmented point clouds) as being in an object database, and the algorithms presented here were used on unrecognized objects. The robot was then asked to move 15 objects in groups of 3, autonomously, from one side of a table to the other and then back, while avoiding obstacles. The set-up is shown in Figure 1. In this experiment, there were 32 grasp attempts. 19 grasps used open-loop, precomputed grasps of recognized object poses, and 16 of these succeeded; 13 grasps used the algorithms presented here, and all 13 succeeded, showing that the methods discussed can perform well in realistic, unstructured situations.

Videos of the grasp selection process, reactive grasping, and a few of these experiments can be seen at http://www.youtube.com/watch?v=jfpecgVnCMo .

VI. Conclusion

In this paper, we showed that simple heuristics can select stable grasps for a large variety of objects even when only partial object scans are available. We further showed that local reactive behaviors based on tactile sensing can significantly increase the success rate for grasps under positional errors or modeling uncertainty.

The intuition behind the quantitative results that we have presented is that many human-designed objects can be grasped just by starting from either above or to the side of the object, aligning the hand with the object principal axes, and trying to grasp it around the center. If the center is not graspable, any other part that fits inside the hand can be attempted, along similar guidelines. Furthermore, tactile sensing is highly informative for performing adjustments based on object geometry around the attempted grasp point, for cases where the estimate of the object's overall shape or position deviate from initial expectations.

We expect these methods to break down in situations requiring complex, situation-specific grasp execution. These include slippery bottles grasped by the handle; desired grasps that inherently look marginal in terms of their contacts; objects that are too light to be sensed by current tactile sensors; objects too large to fit within the gripper (at which point a two-handed grasp is required); flat objects (which could be either slid off the table or into a dustpan-like tool); and objects that are not easily seen by the sensor generating the object point clouds (transparent, shiny, small and flat, etc). All of these situations are the subject of future work.

References

[1] R Balasubramanian, L Xu, P Brook, J Smith, and Y Matsuoka. Human-Guided Grasp Measures Improve Grasp Robustness on a Physical Robot. In *ICRA*, 2010.

[2] A.M. Dollar, L.P. Jentoft, J.H. Gao, and R.D. Howe. Contact sensing and grasping performance of compliant hands. *Autonomous Robots*, 2010.

[3] J. Felip and A. Morales. Robust sensor-based grasp primitive for a three-finger robot hand. In *IROS*, 2009.

[4] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic. Learning of 2D grasping strategies from box-based 3D object approximations. In *RSS*, 2009.

[5] K. Hsiao, L. P. Kaelbling, and T. Lozano-Perez. Robust belief-based execution of manipulation programs. In *WAFR*, 2008.

[6] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A.Y. Ng. Reactive grasping using optical proximity sensors. In *ICRA*, 2009.

[7] A. Jain and C.C. Kemp. EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces. *Autonomous Robots*, 2010.

[8] R. Platt Jr., A. H. Fagg, and R. A. Grupen. Nullspace composition of control laws for grasping. *IROS*, 2002.

[9] I. Kamon, T. Flash, and S. Edelman. Learning to grasp using visual information. In *IEEE International Conference on Robotics and Automation*, pages 2470–2476. Citeseer, 1996.

[10] Z.C. Marton, L. Goron, R.B. Rusu, and M. Beetz. Reconstruction and Verification of 3D Object Models for Grasping. In *ISRR*, 2009.

[11] Z.C. Marton, R.B. Rusu, D. Jain, U. Klank, and M. Beetz. Probabilistic Categorization of Kitchen Objects in Table Settings with a Composite Sensor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA*, 2009.

[12] B. Mayton, L. LeGrand, and J.R. Smith. An Electric Field Pretouch System for Grasping and Co-Manipulation. In *ICRA*, 2010.

[13] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, pages 1824–1829, 2003.

[14] L. Natale and E. Torres-Jara. A sensitive approach to grasping. In *Proceedings of the Sixth International Workshop on Epigenetic Robotics*, 2006.

[15] R. Pelossof, A. Miller, P. Allen, and T. Jebara. An svm learning approach to robotic grasping. In *International Conference on Robotics and Automation*, 2004.

[16] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *International Conference on Robotics and Automation*, 2006.

[17] M. Prats, P. Martinet, S. Lee, and P.J. Sanz. Compliant physical interaction based on external vision-force control and tactile-force combination. In *MFI*, 2008.

[18] A. Saxena, L. Wong, and A.Y. Ng. Learning grasp strategies with partial shape information. In *AAAI Conf. on Artificial Intelligence*, 2008.

[19] Siddhartha Srinivasa, Dave Ferguson, Mike Vande Weghe, Rosen Diankov, Dmitry Berenson, Casey Helfrich, and Hauke Strasdat. The robotic busboy: Steps towards developing a mobile robotic home assistant. In *10th International Conference on Intelligent Autonomous Systems*, 2008.

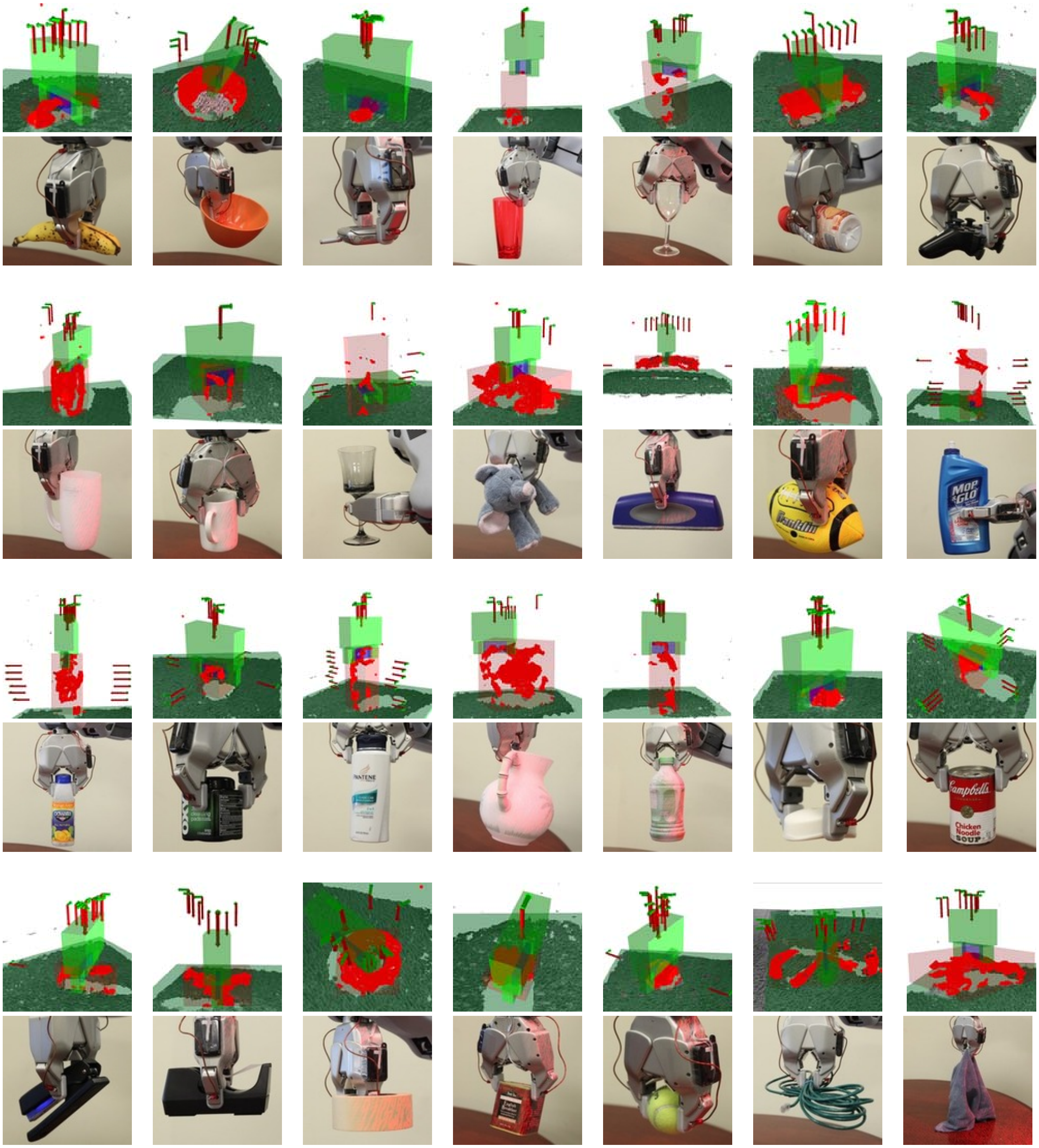

Fig. 9. Grasp experiments: for each object, the top picture shows a visualization of the possible grasps found for the object's point cloud, with the gripper model placed at the highest-ranked grasp, and the transparent box around the object shows the object bounding box. The bottom picture shows one of the resulting grasps for that object.