Semi-Autonomous Manipulation of Natural Objects

Dov Katz, Moslem Kazemi, J. Andrew Bagnell and Anthony Stentz

CMU-RI-TR-12-33

November 2012

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

© Carnegie Mellon University

Abstract

Effective deployment of robots in search and rescue missions will enable faster and safer removal of debris and other hazardous material. As a result, additional lives will be saved, the number and severity of injuries will decrease, and significant damage to infrastructure will be avoided. Already today robots are an integral part of many search and rescue units. These robots typically serve as either mobile cameras (autonomy in navigation, but no manipulation capabilities) or as tools controlled by a human operator (no autonomy, but capable of interacting with the environment). We propose a robotic manipulator that shares a role with the human operator: the robot provides the operator with processed visual information and a set of possible actions, and the human operator chooses the desired next interaction with the environment. To that end, we develop a novel scene segmentation algorithm based on 3D data and a toolbox of compliant motion controllers. We evaluate our approach in real-world experiments in which our robot is tasked with clearing piles of unknown natural objects.

Contents

1 Introduction **1**

2 Related Work **2**
2.1 Scene Segmentation 2
2.2 Grasping 3

3 System Overview **4**

4 Perception **4**
4.1 Detecting Facets 5
4.2 Extracting Actionable Information 5
4.3 Experimental Evaluation 6

5 Compliant Control **7**
5.1 Compliant Grasping 8
5.2 Compliant Pushing/Pulling 10
5.3 Implementation and Experiments 10

6 Graphical User Interface **11**

7 Experimental Validation **11**

8 Lessons Learned **12**

1 Introduction

Efficient and reliable robotic manipulation of natural objects (Figure 1) in the aftermath of a disaster will increase the number of lives saved, decrease the number and severity of injuries, and have a significant economic impact by limiting additional damage to infrastructure and equipment.

Already today, a variety of machines participate in search-and-rescue efforts by lifting heavy objects, manipulating in dangerous environments, and handling hazardous material such as nuclear waste or explosives. These machines provide the necessary power and safety, but are essentially remot controlled tools operated by experts on site. They require a human expert to control every aspect of mobility and interaction with the environment. Because of the high bandwidth requirements of control, the operator must be on site, usually inside or next to the machine, an undesired, if not impossible, requirement.

A straight forward approach to removing the on-site presence requirement is teleoperating a robot from an off-site location [17]. However, the real-time and bandwidth requirements associated with controlling a robot during its interaction with the environment while considering force and visual feedback are prohibitive. An alternative approach, on the other end of the spectrum, strives to develop autonomous navigation and manipulation capabilities. Despite its promise, truly autonomous robots will probably remain out of reach for some years to come. This is because the current state of the art in perception, planning, and AI still cannot support autonomous manipulation even in simpler and well structured environments such as our homes and offices [16].

In this article we explore a third approach: sliding autonomy. In this approach, the robot and the human operator share a role [15]. The degree of autonomy is determined based on the capabilities of the robot and the specifics of the task. This approach benefits from the best of both worlds: On one hand, we let the robot react in real-time to changes in the environment without completely depending on the operator. And on the other hand, we let the operator provide input from a remote and therefore safe location, therefore reducing the cognitive load on the robot.

A fundamental requirement for our role-sharing approach is efficient communication between the human operator and the robot. Both the robot and the operator must be able to refer to the same objects in the environment and understand what interactions are feasible. The operator must be able to specify the next action as loosely as possible, and the robot must be able to translate these instructions into a concrete motion plan.

To enable efficient communication, we have developed a system that is composed of a graphical user interface, a perception library and a set of compliant controllers. Our system presents the human operator with a 3D view of the scene and a segmentation of the scene into interesting objects. The interface also provides the operator with a set of feasible action (pushing, pulling, and grasping) that can be applied to each detected object. The operator can use the visual information to infer the correctness of the proposed segmentation. It can then task the robot with a pair of action and target object. The robot in turn instantiates the appropriate controller based on the selected action and the properties of the selected object.

In the following we describe the development of this system. First, we describe our main contribution: a novel segmentation algorithm that leverages geometric in-

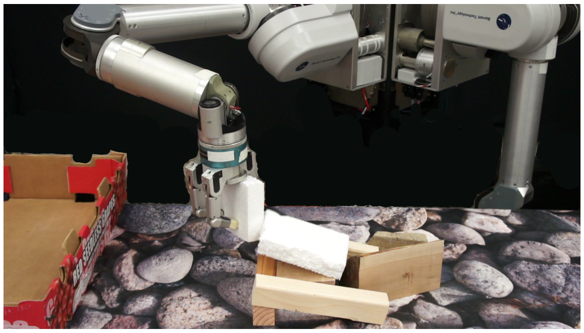


Figure 1: Andy (DARPA's ARM-S platform) manipulating a clutter of natural objects taken from a construction site.

formation to segment a scene into interesting regions. We refer to these regions as "facets". These facets typically correspond to the faces of objects, and are reliable for a large range of object geometries, sizes, and texture. Second, we describe a library of compliant grasping and manipulation controllers that are parameterized by the facets information on which they act. And finally, we describe a novel graphical user interface enabling a human operator to communicate with a robot in low bandwidth. The second contribution of this work is in integrating together perception capabilities, a compliant control library, and a user interface. We validate the merits of the integrated system, as well as the performance of the individual components, in real-world experiments with a robotic manipulator and a set of natural objects (debris taken from a construction site). Our experiments demonstrate the effectiveness of our approach in the task of clearing a cluttered environment composed of these natural objects.

2 Related Work

In our proposed system, the process of manipulating unknown objects has three steps: first, the robot perceives the scene and segments it into regions of interest. Second, a person communicates with the robot via a graphical user interface to indicate the best next interaction. And finally, the robot execute the interaction by instantiating one of the available controllers. We now discuss relevant work to the first and last part: scene segmentation and object grasping.

2.1 Scene Segmentation

Segmentation algorithms [7, 28] divide an entire image into spatially contiguous regions that share a particular property. These methods process an image to identify boundaries between regions that share a particular property. All of the methods in this category are based on the assumption that the boundaries of objects correspond to discontinuities in color, texture, or brightness—and that these discontinuities do not occur anywhere else. Most methods rely on thresholding, edge detection, clustering, or region growing to group pixels based on brightness, color, or texture [7].

The fundamental assumption underlying these methods—namely, that discontinuities of an image property indicates object boundaries—does not hold when multiple objects are present. As clutter increases, objects are more likely to touch each other and many of the visual discontinuities disappear. Also, in scenes that include natural objects and debris, many objects are visually similar. As a result, the boundaries determined by the above algorithms would rarely coincide with the actual object boundaries.

Another category of segmentation algorithms analyzes sequences of images in which objects move relative to each other. This can be accomplished using optical flow [29], statistical methods [6, 19, 22, 23], wavelet transforms [12, 25], and factorization methods [5, 8, 26]. More recently, interactive segmentation algorithms [11, 13] were proposed in which the robot interacts with the environment to create object motion in service of segmentation. However, while relative motion is an important cue for segmentation, it is also costly. In practice, it may be undesirable and even dangerous to stir a pile to generate relative motion.

The perceptual skill we propose belongs to a third category of segmentation algorithm: geometry based segmentation [24,27]. Methods in this category exploit geometric information to extract contiguous regions and to determine the boundaries between those regions. These methods enjoy an increase in popularity due to the recent introduction of cheap, off-the-shelf RGB-D sensors. Most of the geometric segmentation methods are parametric methods—they fit to the data a set of predetermined shapes such as spheres, cylinders, and most frequently planes. In practice, it is difficult to fit these geometric shapes to debris. Thus, we propose a non-parametric method. We too leverage geometric information. However, very much like intensity based segmentation methods, we extract object boundaries based on discontinuities in depth and surface normal orientation.

2.2 Grasping

Robotic grasping is a very well studied field. The majority of the literature on grasping assumes that an a priori model of the object to be manipulated is available. Methods in this category treat grasping as a purely geometric problem. They introduce grasp quality metrics and use them to synthesize grasps in 2D or 3D [3].

Within the category of methods that do not assume prior knowledge, there are two major trends. The first view separates grasping into two stages: acquiring an object model and planning. For example, Hauck et al. use a stereo-vision system to detect and triangulate grasping points on the silhouette of an object [9]. Morales et al. assume planar extruded objects, for which two- and three-fingered grasps are planned based on the object's detected contour [14]. And Saxena et al. propose a classifier for detecting grasp points in images [21]. The classifier is learned from a set of labeled training examples. The resulting classifier usually prefers pairs of edge-like features, ignoring any depth information.

The second view integrates the two steps into a single process. Here, grasping hypotheses are continuously updated by integrating sensor measurements as they become available. For example, Calli et al. use an eye-in-hand system and apply a visual servoing scheme that maximizes the curvature of the object silhouette, thus leading the hand to concave parts of the object [4]. And, Platt et al. use tactile feedback to refine a grasp after initial contact by controlling two opposing fingers along the object surface [18]. They show that their strategy converges to force-closure for arbitrary convex objects.

All of the above methods are designed for man-made objects. Typically, grasping of such objects is only a first step towards using the object. This poses specific requirements on how the object should be grasped (e.g. grasping a cup by its handle is good, grasping a knife at the tip of the blade is bad). In this paper, our focus is on grasping and manipulating natural objects. Our main objective is to remove these objects from a pile. Thus, grasping does not have to be very precise. We leverage this insight and contend that the expensive process of acquiring high quality object models for grasping can be avoided.

Interestingly, recent studies show that humans too often rely on a simple metric to predict the quality of grasp: the orthogonality of the wrist orientation relative to the objects principal axis [2]. This approach requires perception to only estimate the boundaries of an object, and compute its principal components—a much simpler task than shape estimation.

Eliminating the requirements from perception does not come for free. In place of accurate models from perception, we now require controllers that can compensate for these inaccuracies. Kazemi et al. recently proposed a set of compliant grasping primitives which leverage compliant contact with the environment to account for inaccuracies in modeling and localization [10]. The proposed controllers rely on minimal information about the object (e.g., center of gravity and principal axis) to generate and execute a grasp. These controllers are safe and well suited for grasping in clutter due to their compliant nature. Inspired by [10], we devised and implemented a set of compliant grasping controllers which rely on minimal information acquired for each facet through our perception system (see Section 5).

3 System Overview

Our proposed system for manipulating unknown natural objects in clutter is composed of three components: perception, control, and a graphical user interface. The perceptual component is responsible for segmenting an unknown scene into a set of regions that correspond to object or object parts. In addition, perception extracts information about each such region. This information is then used by the second component: control. Our system can instantiate one of the available controllers with the information acquired from perception. The decision which controller to apply to what object is performed by a human operator via the third component: a graphical user interface. We know describe each of the three components in detail.

4 Perception

To perceive an unknown scene composed of natural objects such as rocks, wood and other debris, we developed a novel segmentation algorithm. Our algorithm identifies contiguous regions in RGB-D sensor data by extracting two types of discontinuities:

depth discontinuities and abrupt changes in surface normal orientation. We refer to the segmented regions as "facets", as each region typically corresponds to a side of an object.

4.1 Detecting Facets

Detecting facets from a scene is composed of the following three steps: computing depth discontinuities, estimating surface normals, and image segmentation. This process is illustrated in Figure 2.

To compute **depth discontinuities**, we convolve the depth image with a non-linear filter. This filter computes the maximal depth change from every pixel to its immediate 8 neighbors. If this distance is larger than a pre-defined threshold (in our case, $2cm$), the pixel is marked as a depth discontinuity. See Fig. 2(b) for an example.

To estimate the **surface normal** at any point of the 3D point cloud we fit a local surface to the neighborhood of the point, tangent to the surface. Then, we compute a normal to that plane. Thus, we solve a least-square plane fitting. This can be done by analyzing the principal components of a covariance matrix created from the nearest neighbors of the point. The matrix is computed as

$$C = \frac{1}{k} \sum_{i=1}^{k} (p_i - \bar{p}) \cdot (p_i - \bar{p})^T$$

and v_j satisfies

$$C \cdot v_j = \lambda_j \cdot v_j, j \in \{0, 1, 2\}$$

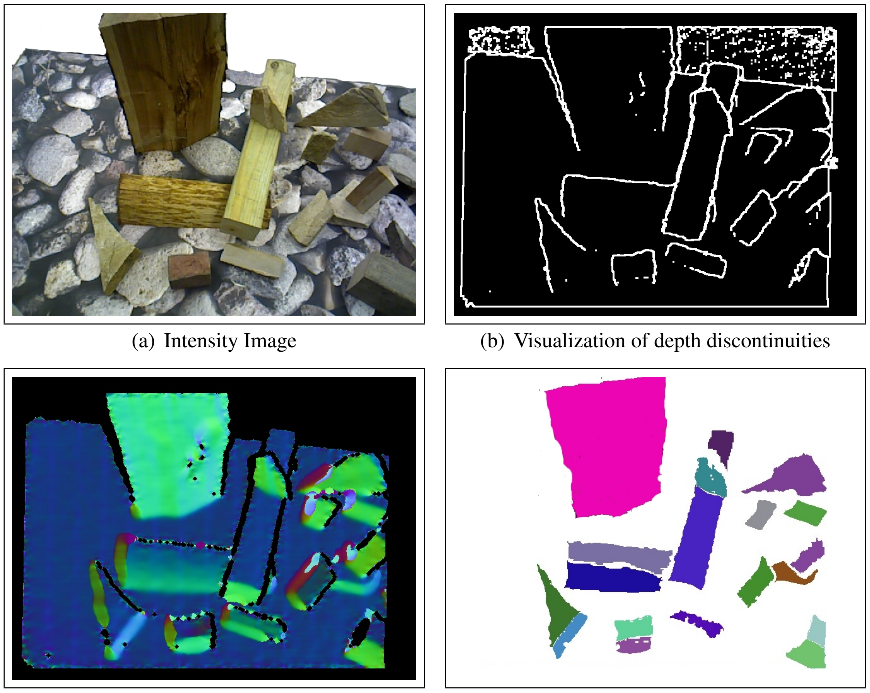
, where k is the number of points considered in the neighborhood of p_i, and $\bar{p}$ represents the 3D centroid of the set of k nearest neighbors. λ_j is the j-th eigenvalue of the covariance matrix, and v_j is the j-th eigenvector. Our implementation is based on the opensource Point Cloud Library (PCL [20]). We visualize the computed normals as an intensity image, in which each channel (R, G, and B) corresponds to a direction of the normal (X, Y, and Z). See Fig. 2(c) for an example.

Finally, we **extract facets** in the scene by overlaying the depth discontinuities over the surface normals, thereby obtaining a color image in which both abrupt depth changes and normal orientation changes can be detected. We can now treat the problem as a classical image segmentation. We use OpenCV's meanshift segmentation algorithm. The result of facet detection are illustrated in Fig. 2(d).

4.2 Extracting Actionable Information

To complete the perception component of our system, we now analyze the detected facets. For each facet, we extract information that is necessary to parameterize our compliant controllers: center of gravity (COG) and principal axis. Together, perception provides the human operator with the set of detected objects and a set of actions that can be applied to each object.

Estimating the center of gravity is a simple matter of averaging the 3D point cloud. And, estimating the principal axis of a facet is achieved by computing principal components analysis (PCA) on the corresponding point cloud. The center of gravity indicates



(a) Intensity Image

(b) Visualization of depth discontinuities

(c) Visualization of surface normals. Intensity values (RGB) correspond to the direction of surface normals (XYZ).

(d) Extracted Facets

Figure 2: Facet detection algorithm: depth discontinuities and surface normals discontinuities are used to determine the boundaries between image segments.

the desired position of the palm, and the principal axis indicates the orientation of the wrist for grasping or pulling/pushing. Our approach makes two assumptions: the density of a facet is uniformly distributed and the entire facet is visible to the robot. In practice, both assumption are frequently violated. Nevertheless, it usually provides us with a good enough guess. Figure 3 shows an example of detecting centers of gravity and principal axes.

4.3 Experimental Evaluation

To evaluate the performance of our perception, we conducted a series of experiments with four different objects (see Fig. 4). The objects vary in size, shape, appearance, and material. In every experiment, we placed an object on a visually confusing background (a poster of gravel) in five different configurations. We then extracted the detected facets. The odd rows in Fig. 4 show the objects, and the even rows show the corresponding detected facets.

In all cases the object is detected correctly. The facet corresponds well to the physical boundaries of the object. In the last configuration of the second object (4^{th} row,

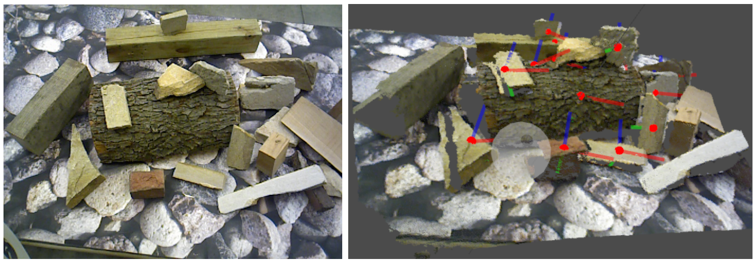


Figure 3: Extracting actionable information from facets. For the detected facets, we compute the COG (red circle) and principal axes (axis are color coded: red = principal axis, green = secondary axis, blue = trinary axis).

5^{th} column) only one object facet was detected. This is not a failure. The second facet that was detected in the other configurations is simply not visible here. In most configurations of the last experiment (5^{th} and 6^{th} row) only one large facet is detected. This is because the remaining object facets are too small for the resolution of our sensor.

Figure 5 shows for each of the four objects an example of the detected center of gravity and principal axes. This information is computed based on the detected facets. We rely on this information to parameterize our compliant controllers. In all cases the detected center of gravity corresponds to the approximate center of the facet, and the axes are aligned with the object.

5 Compliant Control

When accurate models are available and precise localization is achievable, a motion plan is best executed using stiff controllers that can guarantee the execution of a trajectory. However, in the presence of modeling and localization uncertainties, rigidly following a trajectory becomes dangerous for the robot and the environment. Compliant controllers overcome modeling and localization uncertainties by maintaining proper contact with the environment. During the robot's interaction with the environment, a compliant controller responds to the detected contact forces.

Inspired by the work of Kazemi et al. [10], we devised compliant controllers to manipulate unknown natural objects in clutter. These controllers are instantiated using minimal information about the target object. In our case, we only require an estimate of the object's center of gravity and principal axis. The compliant motion primitives effectively address the modeling and localization uncertainties by maintaining proper contacts with objects and support surfaces during execution.

Our compliant motion primitives are velocity-based operational space controllers (see [10] for details). They rely on force feedback acquired by a force-torque sensor installed at the robot's wrist. The fingers' motion is also coordinated and controlled using position-based controllers during grasp execution. We devised two compliant motion primitives to manipulate unknown natural objects: compliant grasping and compliant pulling/pushing primitives.

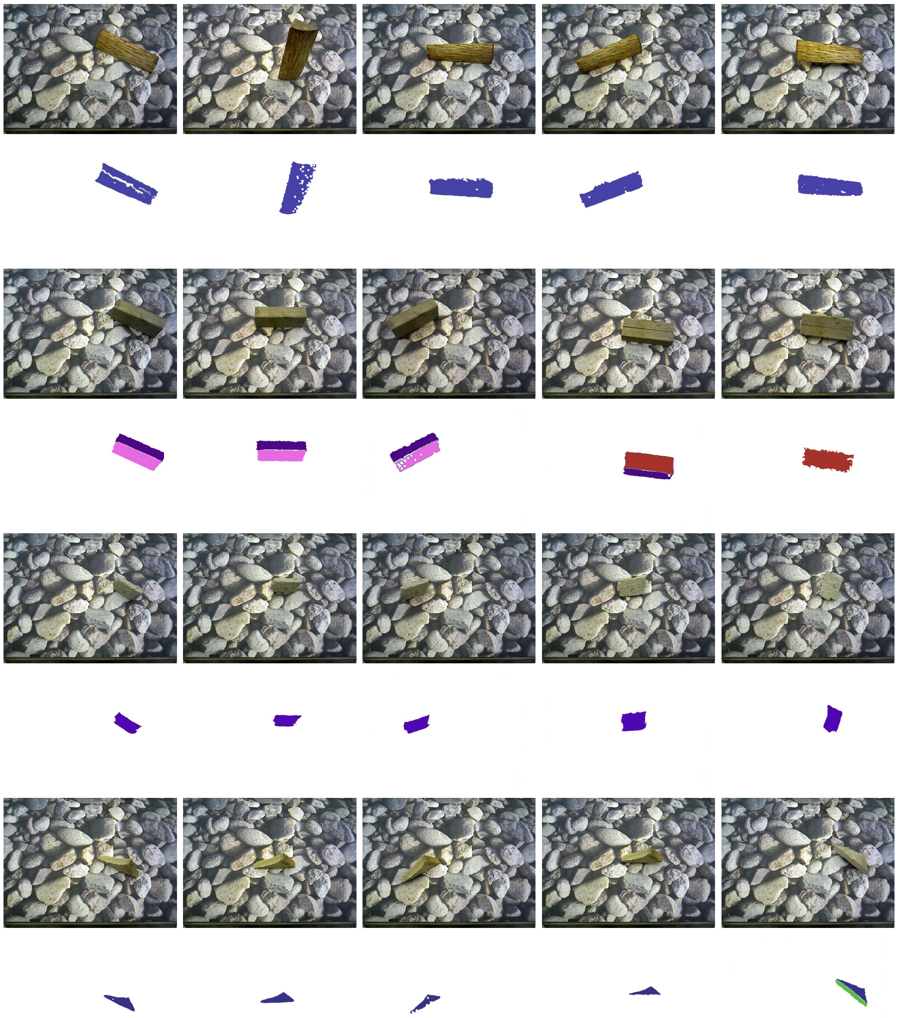


Figure 4: Experimental evaluation of facet detection. Odd rows show an object in 5 configurations. Even rows show the extracted facets. All visible facets that are large enough for the resolution of the sensor were detected successfully.

5.1 Compliant Grasping

We define a grasp by the hand's pre-shape and its launch pose (the pose from which the grasp is to be executed). In this paper, we use a single pre-shape: a cup-like hand pre-shape (Fig.6). Future work could consider other pre-shapes which can be determined using additional information about the shape of the object.

The configuration of the fingers depends on the width of the object. Instead of

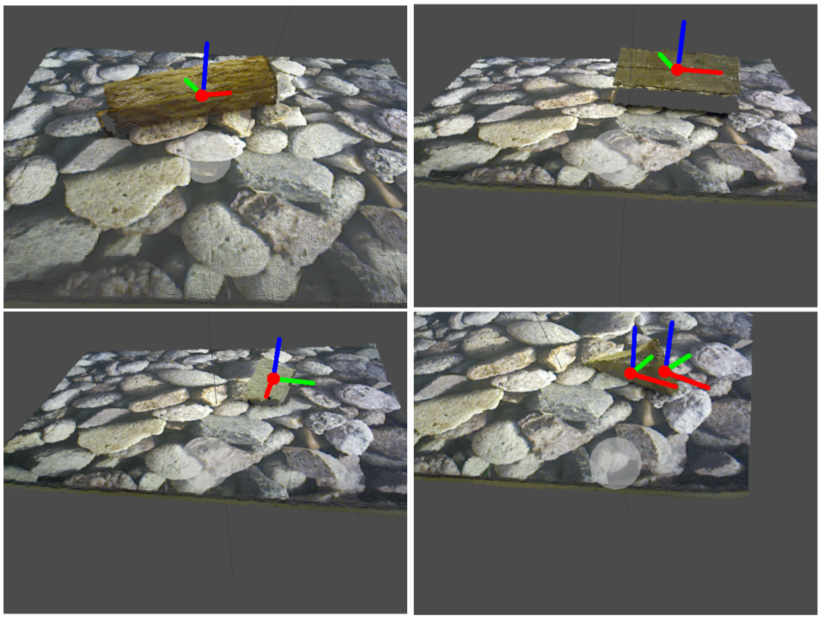


Figure 5: Experimental evaluation of action generation based on the detected facets. Each column corresponds to a single object configuration of the 4 objects in Figure 4. The 3D view shows the object. The center of gravity is marked by a red sphere. The object's axes are color coded: principal axis (red), secondary axis (green), and trinary axis (blue). In all cases the position of the center of gravity and orientation of the axes correspond well with the object.

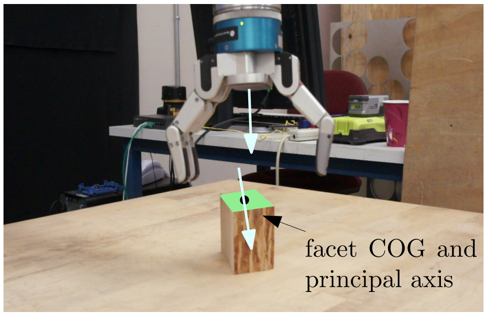


Figure 6: The Barrett hand in a cup pre-shape is positioned above the center of gravity of the facet (marked in green), and alligned with respect to the facet's prinicpal axis.

computing this information from perception, we leverage the compliance of our motion primitives. The fingers are maximally opened before grasping. As soon as the fingertips touch the support surface, they safely close towards the objects, until the object is caged. The grasp launch pose is selected to be at a safe distance above the

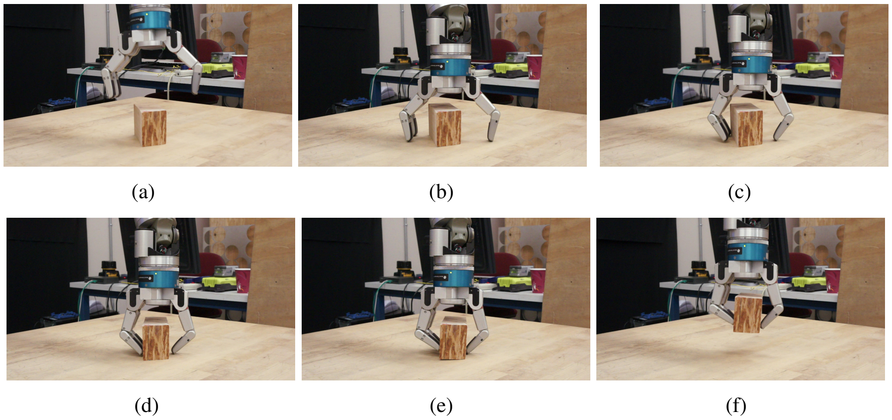

(a) (b) (c) (d) (e) (f)

Figure 7: Compliant grasping of an object from a table top

support surface and the object's COG, and with the palm of the hand parallel to the supporting surface. The operator can specify the support surface normal by selecting three representative surface points.

A compliant grasp is executed as follows: First, we servo the hand along the palm's normal, until contact is detected between the fingertips and the support surface or the object (Fig.7(b)). Second, the fingers close along their pre-defined trajectories while the hand is simultaneously servo controlled (up or down) in compliance with the forces measured at the wrist in a closed-loop fashion to ensure safe and proper contact between the fingertips and the support surface (see Fig.7). We note that the aim of the compliant grasping strategy is not to place the fingertips on certain points on the object surface but to achieve rough, yet robust, grasps of an unknown object.

5.2 Compliant Pushing/Pulling

If an object is surrounded by clutter, preventing the robot from grasping it, a compliant push or pull controller can be executed. The launch pose for the push/pull primitive is calculated similarly to the compliant grasping controller. However, the push/pull action is executed by servoing the hand toward (pull) or away from (push) the robot and in parallel to the support surface. Please note that the hand may not be perfectly parallel to the support surface due to localization uncertainties. Hence, we introduce a compliant motion along the palm normal to servo the hand in compliance with forces measured at the wrist, if the fingertips touch the environment during the push/pull action. The cup-like pre-shape helps to secure the object during the hand movement and avoids missing it.

5.3 Implementation and Experiments

We have implemented the above primitives on a robotic manipulator consisting of a 7-DOF Barrett Whole Arm Manipulator (WAM) and a 3-fingered Barrett hand. Given a launch pose for grasping an object, the system generates a feasible plan to move the WAM to the desired launch pose while avoiding obstacles. After reaching the launch pose the compliant grasping primitive is executed as explained above. The grabbed object is then transferred to a pre-defined target location. After a push/pull action the arm is moved out of the way so the robot's view is not obstructed.

We performed grasp repeatability experiments on 5 different natural objects (similar to the ones in Fig. 4), 5 grasps for each object at fixed location and orientation. Our compliant grasping strategy was successful in 24 out of 25 grasps. The one failure was due to in-hand slippage after performing the grasp and during transport to the target location. We believe that having compliant fingers will help to eliminate such failures. In addition, compliant fingers will also allow the robot to manipulate the clutter safely without the risk of damaging the hardware.

6 Graphical User Interface

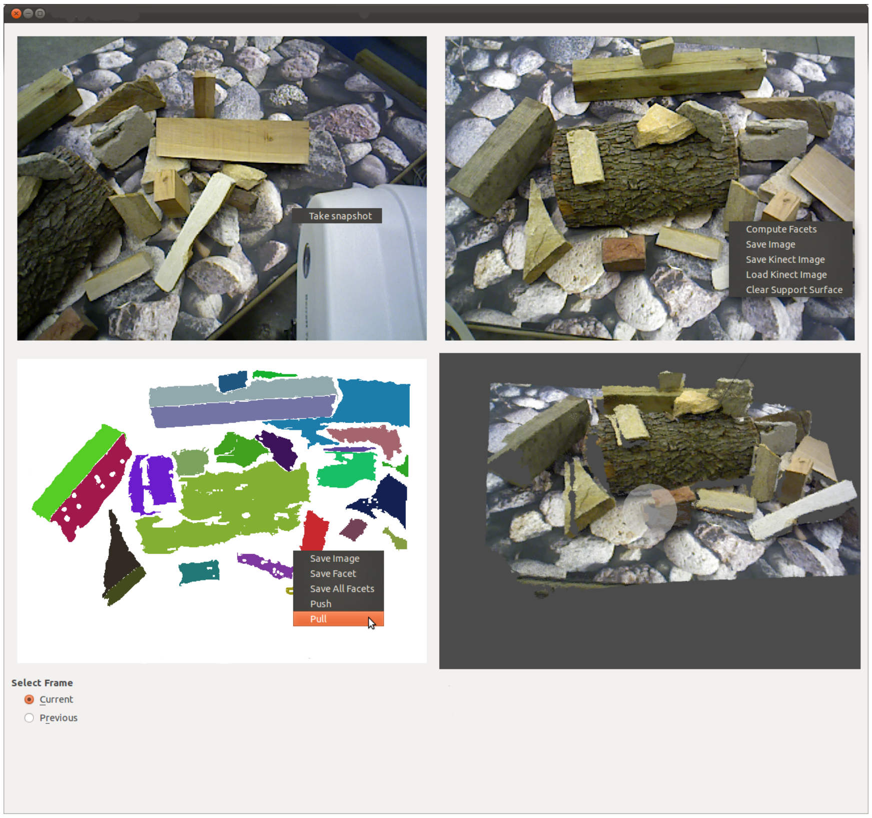
The graphical user interface (Figure 8) provides the human operator with visual information in the form of continuous 2D and 3D view of the scene. The operator can request a snapshot of the scene, which is then analyzed by perception to extract facets and the associated actions. Finally, it enables the operator to select a facet and an action and instruct the robot to instantiate and apply the appropriate compliant controller to the facet.

The communication between the robot (RGB-D sensor and manipulator) and the human operator relies on ROS (Robot Operating System) for communication. ROS requires a simple network connection, and therefore allows the operator to be off-site. Future research could attempt to reduce or even eliminate the need for human supervision. At the moment, the operator guarantees recovery from perception errors and provides dependable decision making.

7 Experimental Validation

To evaluate our system we conducted dozens of experiments with the task of clearing a cluttered table. We placed a textured background poster to challenge the perception component, and randomly configure the scene to include many different real-world objects from a construction site. Experiments were conducted on the DARPA ARM-S robotic manipulation system [1].

Figure 9 illustrates one sequence of interactions to clear a cluttered scene. The sequence begins with a set of objects placed next to each other in an arbitrary configuration. The robot and the human operator collaborate to achieve the task. The sequence shows the intensity image displayed to the operator alongside the detected facets. The human operator selects an action, and the next row shows the scene after the action was applied. We conducted multiple similar experiments to demonstrate the reliability and robustness of the different components (perception, control, and user interface). The results are very promising. In all experiments the human operator is able to manage the operation smoothly and achieve efficient clearing of the debris.



Select Frame
● Current
○ Previous

Figure 8: The Graphical User Interface. **Top left:** Live video. The operator can choose when to take a snapshot of the scene using a context menu. **Top right:** The captured snapshot. It is updated when a new snapshot is requested. The interface can show either the current or previous snapshot (radio button on bottom left). **Bottom left:** The computed facets. The operator can choose from a list of available action by right-clicking on each facet. **Bottom right:** A live 3D display.

The sequence shown in Figure 9 is composed of one pulling action (between the first and second row), and 7 grasping actions. The execution time was 35 seconds for the pull action, and 50, 54, 78, 60, 68, 51, and 55 seconds for the sequence of grasping. On average, it takes about 56 seconds per action. These times include perception, interaction with the user interface, planning a collision free trajectory, grasping an object, dropping it in a container and returning to the home configuration.

8 Lessons Learned

All three components of the proposed system appear to be very promising. The perceptual component for extracting 3D object facets works well despite the challenges

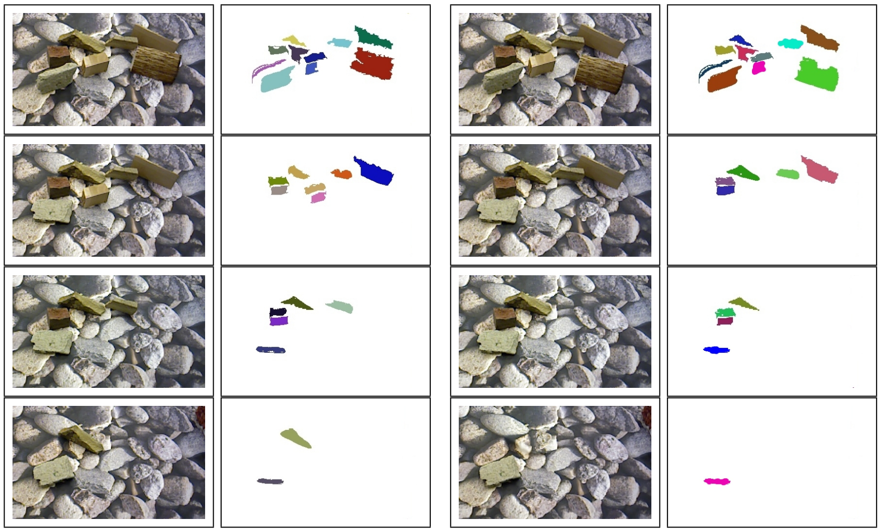


Figure 9: A sequence of interactions to clear a cluttered set of natural objects. Left: intensity image acquired by the sensor. Right: the corresponding segmentation into facets. The sequence ends when no more facets are detected. Videos are available at http://www.dubikatz.com/natural.html

associated with natural objects of arbitrary size, shape, and appearance. Our compliant controllers handle well the inherent uncertainties and are safe and well suited for the task of clearing clutter. And finally the graphical user interface is functional and convenient to use.

There are a few limitations to our approach. First, the user interface does not allow the human operator to specify actions that are not offered by the robot. In the case where the robot's perception fails to detect an object (e.g. the end of the sequence in Fig. 9), it would be good to let the operator specify an action. Second, the hardware we use is not ideal for this type of tasks. We depend on the manipulator being compliant, powerful, and durable. Our Barrett arm and hand are somewhat compliant. However, we found that most rocks that fit in the hand are simply too heavy for the mechanism. Also, this is an expensive hardware. It is not well protected against scratches and damage from bumping into rough objects such as bricks, rocks, and other debris. In future work, we intend to look into alternative hardwares. We will also continue to expand the set of perceptual capabilities and actions that are identified by the robot and provided for the human operator's consideration.

ACKNOWLEDGMENT

This work was conducted (in part) through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016. The authors also gratefully acknowledge funding under the DARPA Autonomous Robotic Manipulation Software Track (ARM-S) program.

References

[1] J. A. Bagnell, F. Cavalcanti, L. Cui, T. Galluzzo, M. Hebert, M. Kazemi, M. Klingensmith, J. Libby, T. Y. Liu, N. Pollard, M. Pivtoraiko, J.-S. Valois, and R. Zhu. System design and implementation for autonomous robotic manipulation. In *IROS*, 2012.

[2] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *2010 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2294–2301, 2010.

[3] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 348–353, San Francisco, CA, USA, 2000.

[4] B. Calli, M. Wisse, and P. Jonker. Grasping of unknown objects via curvature maximization using active vision. In *2011 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[5] J. Costeira and T. Kanade. A Multibody Factorization Method for Independently Moving Objects. *International Journal of Computer Vision*, 29(3):159–179, September 1998.

[6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer Segmentation of Live Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 53–60, Washington, DC, USA, 2006. IEEE Computer Society.

[7] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[8] A. Goh and R. Vidal. Segmenting Motions of Different Types by Unsupervised Manifold Clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, June 2007. IEEE Computer Society.

[9] A. Hauck, J. Ruttinger, M. Sorg, and G. Farber. Visual determination of 3D grasping points on unknown objects with a binocular camera system. In *1999 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, 1999. IROS '99. Proceedings*, volume 1, pages 272–278 vol.1, 1999.

[10] M. Kazemi, J.-S. Valois, J. A. Bagnell, and N. Pollard. Robust object grasping using force compliant motion primitives. In *Robotics: Science and Systems*, July 2012.

[11] J. Kenney, T. Buckley, and O. Brock. Interactive Segmentation for Manipulation in Unstructured Environments. In *ICRA*, pages 1343–1348, Kobe, Japan, May 2009. IEEE Press.

[12] M. Kong, J.-P. Leduc, B. K. Ghosh, and M. V. Wickerhauser. Spatio-Temporal Continuous Wavelet Transforms for Motion-Based Segmentation in Real Image Sequences. In *International Conference on Image Processing*, volume 2, pages 662–666, Chicago, Illinois, USA, October 1998. IEEE Computer Society.

[13] G. Metta and P. Fitzpatrick. Early Integration of Vision and Manipulation. *Adaptive Behavior*, 11(2):109–128, June 2003.

[14] A. Morales, P. J. Sanz, A. P. del Pobil, and A. H. Fagg. Vision-based three-finger grasp synthesis constrained by hand geometry. *Robotics and Autonomous Systems*, 54(6):496–512, 2006.

[15] R. Murphy and J. Burke. From remote tool to shared roles. *Robotics Automation Magazine, IEEE*, 15(4):39 –49, dec. 2008.

[16] R. R. Murphy. Humans, robots, rubble, and research. *interactions*, 12(2):37–39, Mar. 2005.

[17] R. R. Murphy and J. L. Burke. Up from the rubble: Lessons learned about hri from search and rescue. In *in Proceedings of the 49th Annual Meetings of the Human Factors and Ergonomics Society. 2005*, pages 437–2005, 2005.

[18] R. Platt, A. H. Fagg, and R. A. Grupen. Null-space grasp control: theory and experiments. *Robotics, IEEE Transactions on*, 26(2):282295, 2010.

[19] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Particle Filtering for Geometric Active Contours with Application to Tracking Moving and Deforming Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–9, Washington, DC, USA, 2005. IEEE Computer Society.

[20] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[21] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *The Intl. Journal of Robotics Research*, 27(2):157, 2008.

[22] H. Shen, L. Zhang, B. Huang, and P. Li. A MAP Approach for Joint Motion Estimation, Segmentation, and Super Resolution. In *IEEE Transactions on Image Processing*, volume 16, pages 479–490. IEEE, 2007.

[23] R. Stolkin, A. Greig, M. Hodgetts, and J. Gilby. An EM/E-MRF Algorithm for Adaptive Model Based Tracking in Extremely Poor Visibility. *Image and Vision Computing*, 26(4):480–495, 2008.

[24] C. J. Taylor and A. Cowley. Segmentation and analysis of rgb-d data. In *RSS 2011 Workshop on RGB-D Cameras*, June 2011.

[25] L. Wiskott. Segmentation from Motion: Combining Gabor and Mallat-Wavelets to Overcome the Aperture and Correspondence Problems. *Pattern Recognition*, 32(32):1751–1766, 1999.

[26] J. Yan and M. Pollefeys. Automatic Kinematic Chain Building from Feature Trajectories of Articulated Objects. In *CVPR*, pages 712–719, USA, 2006.

[27] S.-W. Yang, C.-C. Wang, and C.-H. Chang. Ransac matching: Simultaneous registration and segmentation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1905 –1912, may 2010.

[28] L. Zappella. Motion Segmentation from Feature Trajectories. Master's thesis, University of Girona, Girona, Spain, 2008.

[29] J. Zhang, F. Shi, J. Wang, and Y. Liu. 3D Motion Segmentation from Straight-Line Optical Flow. In *Multimedia Content Analysis and Mining*, pages 85–94. Springer Berlin / Heidelberg, 2007.