2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Clearing a Pile of Unknown Objects using Interactive Perception

Dov Katz, Moslem Kazemi, J. Andrew Bagnell and Anthony Stentz[1]

***Abstract*— We address the problem of clearing a pile of unknown objects using an autonomous interactive perception approach. Our robot hypothesizes the boundaries of objects in a pile of unknown objects (object segmentation) and verifies its hypotheses (object detection) using deliberate interactions. To guarantee the safety of the robot and the environment, we use compliant motion primitives for poking and grasping. Every verified segmentation hypothesis can be used to parameterize a compliant controller for manipulation or grasping. The robot alternates between poking actions to verify its segmentation and grasping actions to remove objects from the pile. We demonstrate our method with a robotic manipulator. We evaluate our approach with real-world experiments of clearing cluttered scenes composed of unknown objects.**

I. Introduction

Autonomous manipulation of unknown objects in a pile (Fig. 1) is a prerequisite for a large variety of robotic applications ranging from household robotics to flexible manufacturing and from space exploration to search and rescue missions. In this work, we address the problem of removing unknown objects from a pile. This is an important task as it enables necessary capabilities such as object counting, arranging, and sorting.

Manipulating a pile of unknown objects is challenging because it requires close integration of multiple capabilities, including perception, manipulation, grasping, and motion planning. Moreover, because of the complexity associated with perceiving and interacting with a pile of unknown objects, each of these capabilities can easily fail: Object recognition may fail due to occlusion by other objects in the pile or difficulty to determine object boundaries. Grasping can fail when object recognition fails, resulting in an attempt to grasp at the wrong location or grasping multiple objects simultaneously. And motion planning is particularly prone to error when moving in an unknown cluttered environment. Also motion execution itself must be careful to avoid damage to the robot or the environment.

To address the above challenges, we propose an interactive perception approach in which the robot can actively verify its understanding of the pile. Our robot segments a scene into a set of object hypotheses. Next, the robot interacts with the environment to verify the correctness of its segmentation hypotheses. A verified hypothesis corresponds to an object's facet, and is used to parameterize a compliant grasping controller. After successfully grasping an object, the robot removes it from the pile and release the object into a container. This process continues until all objects have been removed from the pile.

[1]The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Fig. 1: Perceiving and manipulating unknown objects in a pile. Top: Our robot Andy (DARPA's ARM-S platform). Bottom: A typical pile of unknown (manmade and natural) objects used in our experiments.

The two main contributions of this work are: (1) the development of an interactive segmentation and segmentation-verification algorithm for manipulating unknown objects, and (2) the integration of all aspects of perception, manipulation, grasping, and motion planning into a single system. Our system is fully autonomous: the robot segments an object, interacts with it to verify that segmentation is correct, and instantiate a compliant controller to either poke or grasp the object. In our current implementation, the robot selects which object to poke or grasp next at random. However, in future work, we intend to explore self-supervised learning of the best next action.

Our method allows for robust, reliable, and safe interaction in unstructured environments because it relies on two pillars: interactive perception [12]–[14] and compliant motion [15]. *Interactive perception* enables the robot to reveal and verify perceptual information. In our case, interaction creates change in the environment, which enables the robot to verify its initial segmentation hypotheses. If the robot fails to verify a segmentation hypothesis, it can simply interact with the environment again. Once a segmentation hypothesis is verified, perception provides reliable information for grasping an object and removing it from the pile. And *compliant motion* enables safe interaction despite the inevitable uncertainty in modeling and localization.

In the following we describe our system for autonomous clearing of a pile of unknown objects. In section II we discuss related work. Then, we provide an overview of our system in section III, followed by detailed discussion of

978-1-4673-5642-8/13/$31.00 ©2013 IEEE

the three main components in sections IV- VI. Finally, we present experimental results demonstrating the robustness of our method in section VII.

II. Related Work

Our algorithm is composed of three main components: an image segmentation algorithm, an object detection algorithm, and compliant poking and grasping primitives. We now discuss relevant work to these three components.

A. Scene Segmentation

Segmentation algorithms [7], [24] process an image and divide it into spatially contiguous regions that share a particular property. These algorithms assume that boundaries between objects correspond to discontinuities in color, texture, or brightness—and that these discontinuities do not occur anywhere else. In practice, these assumptions are frequently violated. Moreover, most segmentation methods become brittle and unreliable when applied to clutter because of the significant overlap between objects.

A more reliable cue for object segmentation is motion. Segmentation from motion algorithms analyze sequences of images in which objects are in motion. This motion is either assumed to occur [8], [21], [25] or can be induced by the robot [16]. Relative motion is a conclusive clue for object segmentation. However, existing methods only allow planar motion and do not consider occlusion—both of which are unrealistic when interacting with a pile of objects. In contrast, our interactive approach allows general 3D motion and handles occlusion. It is composed of two parts: generating segmentation hypotheses using geometric information and using interaction to verify these hypotheses.

Geometric segmentation algorithms extract geometrically contiguous regions to determine the boundaries between objects [22], [23]. These algorithms rely on depth information acquired by RGB-D sensors. They are typically parametric methods, fitting a set of predetermined shapes such as spheres, cylinders, and most frequently planes to the data. These methods assume that objects can be described using a single shape primitive. In practice, this is rarely the case. Moreover, these methods are unreliable in clutter because objects overlap. We address these limitations with a non-parametric approach. Our algorithm extracts region boundaries based on discontinuities in depth and surface normals orientation.

Without prior knowledge, every segmentation algorithm becomes less reliable in clutter. Also our non-parametric geometric segmentation algorithm can be confused by object overlapping each other. We resolve this limitation using interactive perception. In our approach, segmentation generates hypotheses (object facets) that are verified with interaction. Verified hypotheses are those that were segmented as individual regions before and after the interaction, and have moved as a result of the interaction. This interactive process allows the robot to recover from segmentation errors, therefore increasing the robustness and reliability of our method.

B. Object Detection

Object detection is the task of finding a given object in an image. It can be particularly challenging in the face of changes in perspective, size, or scale, and when the object is partially obstructed from view. There is an extensive body of work in computer vision about object detection (or: object recognition) [7]. If an a priori CAD model of the target object is available, edge detection or primal sketches can be used to find a match [18]. When multiple images of an object are available, they can be used as templates to find the closet match [1]. The most important limitation of methods that rely on a priori models is that in unstructured environments such as our homes and offices, those models are unlikely to be available for all objects.

An alternative to model based object detection employs a sparse object representation using key-points such as SIFT features [17]. Object detection requires extracting key-points from two images (template and target), and computing pairwise matching to determine whether the template appear in the target image. Object detection using SIFT features requires a priori template images of individual objects. Our algorithm generates templates on-line: it computes a segmentation of the scene into object facets, and associates SIFT features with each facet. It then evaluates the similarity of two facets (before and after some interaction) by matching their SIFT features. Because SIFT matching alone may not be sufficient (e.g. featureless objects), our method considers additional cues (color, size, and shape) to evaluate the quality of a match. The resulting object detection algorithm is robust to changes in perspective, illumination, and partial occlusions, and it does not require an a priori object model.

C. Grasping

Robotic grasping is very well studied. There is a variety of criteria that one could use to evaluate and guide grasping. For example, the quality of the force/form closure can be used to determine the quality of a grasp [2]. These methods typically assume that an a priori object model is available. If a model is not available, the object can be first modeled by using stereo-vision [10] to extract contact points, by detecting contours [10], or by learning grasp points from labeled images [20]. Then, grasping proceeds based on the acquired model. Alternatively, modeling and grasping can be merged into a single process where grasping hypotheses are continuously updated by integrating sensor measurements as they become available [4], [19].

In this paper, grasping is used for transporting an object from a pile into a predetermined destination (container). We require that grasping is safe to the robot and minimally disruptive to the pile. We guarantee the robot's safety using compliant motion primitives, and our motion planner minimizes collision with other objects. Our grasping and poking primitives are simple. They are instantiated using information extracted from perception: the center of gravity and principal axes of the target facet. This simple approach towards grasping results in reliable interaction (see [15] for detailed discussion).

D. Manipulation in Clutter

Only recently researchers have begun exploring manipulation in clutter. Existing methods such as [5], [11] focus on objects that are planar and move in parallel to the camera. In [9] 3D objects and motions are allowed, but a priori models of all objects in the pile are assumed. In contrast, our method acquires all necessary information from perception, and applies to general 3D objects and 3D motion.

In [5], [9], [11] grasping is performed using a simple parallel jaw gripper. The focus is on singulating objects from the pile to guarantee enough free space around the object. Then, grasping only requires information about the location of the object. In contrast, our method allows grasping from within the clutter. We use the more complex Barrett hand and compliant motion primitives that are instantiated based on the segmented object facets. We consider collision with other objects and the dimensions and configuration of the grasped object to plan the robot's approach and grasp. Because singulation is not necessary, and because grasping is informed by perception, our method is more efficient, requiring an average of 2 interactions per object (poke and grasp), compared to 6.6 interactions per object in [5].

III. Algorithm Overview

Our algorithm is composed of three components: object segmentation, object detection, and action selection and execution. Object segmentation generates object facets hypotheses. This process is described in section IV. Then, the algorithm selects a candidate facet and interacts with it (poking). This is described in section VI. As a result of the interaction, one or more objects (and therefore facets) have moved. The algorithm now computes a new segmentation and compares it to the original segmentation. In this step, we verify the correctness of segmentation by matching facets hypotheses before and after the interaction. We only consider high probability matches and only those associated with moved objects. This interactive process of verifying the correctness of segmentation is described in section V. Finally, the algorithm selects a verified facet, and a compliant grasp is executed to pick the object and transport it to a predetermined destination, where the object is released. This process continues until no more verified facets are available. Figure 2 illustrates the entire process.

IV. Generating Object Hypotheses

In order to interact with unknown objects, we first generate a segmentation of the scene into hypothesized object facets. A facet is an approximately smooth circumscribed surface. It does not have to be flat (plane). Dividing an object into facets is intuitive and repeatable under changes of perspective, lighting condition, and even partial occlusion. To extract object facets, our algorithm identifies two types of geometric discontinuities: depth discontinuities and abrupt changes in surface normal orientation. A segment (facet) is an image region that lies between those discontinuities. Facet detection is composed of the following three steps: computing depth discontinuities, estimating surface normals, and image segmentation. This process is illustrated in Fig. 3.

We compute **depth discontinuities** by convolving the depth image with a non-linear filter. This filter computes the maximal depth change between every pixel and its immediate 8 neighbors. If this distance is larger than 2cm, the pixel is marked as a depth discontinuity. The 2cm threshold is due to the resolution of our RGB-D sensor (Kinect).

The **surface normal** at every point of the 3D point cloud is estimated by fitting a local plane to the neighborhood of the point. We then compute the normal to that plane using 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 C is computed as

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

where v_j satisfies $C \cdot v_j = \lambda_j \cdot v_j, j \in \{0, 1, 2\}$, 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. Figure 3 provides a visualization of the surface normals. The three angles of every normal are represented using the three color channels (RGB).

Finally, we **extract facets** by overlaying the depth discontinuities over the surface normals. The result is a color image representing both types of geometric discontinuities (depth and surface normal orientation). Now, as Fig. 3 shows, extracting facets is equivalent to extracting contiguous color regions in an image. Therefore, we extract facets using a standard color segmentation algorithm: the mean-shift segmentation algorithm implemented in OpenCV. Please note that mean-shift is applied to a color representation of depth and surface normals, not to the RGB image of the scene.

Figure 4 shows three examples of facet segmentation. For every intensity image (left column), there is a corresponding segmentation in the middle column. The right column shows the corresponding point-cloud, and the red circle and axes mark a potential action. We will discuss how such actions are generated and applied to objects in section VI.

Facet detection has two main limitations. First, our sensor (Kinect) cannot perceive reflective materials. And second, our method is not able to distinguish between two objects that are touching each other and have similar surface normals. This could be solved in future work by considering color, texture, and experience.

V. Verifying Object Hypotheses

Segmentation generates a set of object facets hypotheses. We would like to use such an hypothesis to inform grasping. However, without assuming prior knowledge, object segmentation may not be reliable, particularly so in clutter. For instance, under-segmentation can occur if two objects have similar appearance and touch each other. They may be segmented as a single object.

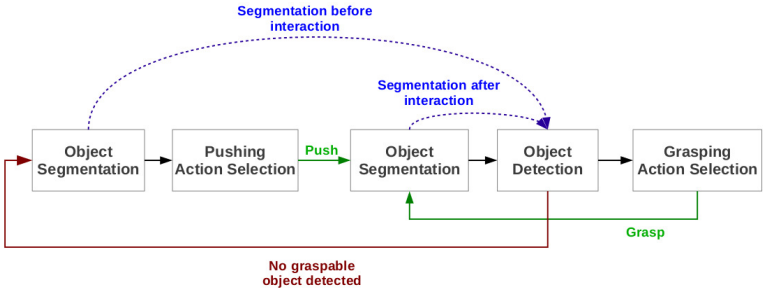


Fig. 2: Algorithm description: the algorithm segments the scene into hypothesized object facets, pokes a facet, verifies segmentation by detecting moved facets that were seen before and after the interaction, and grasps a verified facet. The process continues until all objects have been removed.

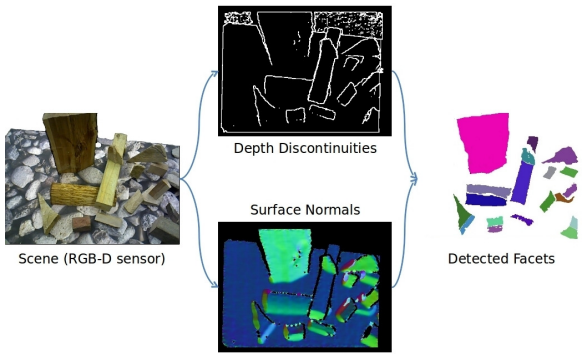


Fig. 3: Facet detection algorithm: The input (left) is an RGB-D image. The algorithm extracts depth discontinuities (top) and normal discontinuities (bottom). The resulting segmentation corresponds to object facets (right).

Fig. 4: Experimental evaluation of facet detection. Left: pile on unknown objects. Middle: segmentation of the scene into facets (color coded). Right: 3D view of the scene. To interact with objects we instantiate compliant controllers with information extracted from each facet: COG (red circle) and principal axes (red = principal axis, green = secondary axis, blue = trinary axis).

Relying on a wrong segmentation to instantiate a grasping controller can be harmful to both the robot and the environment. Under-segmentation may results in an attempt to grasp multiple objects. Consequently, objects may fall and break. And over-segmentation can lead to a wrong parameterization of the controller, resulting in an unreliable grasp. Thus, verifying the correctness of our object hypotheses is crucial.

As visual and geometric information alone may not suffice, our algorithm leverages another strong perceptual cue: motion. We verify the correctness of segmentation using an interactive perception approach in which interaction becomes part of the perceptual process. Our robot interacts with a candidate hypothesis (an object facet) in order to create relative motion. This interaction must be careful and safe. We achieve that with a library of compliant controllers (described in section VI).

As soon as the interaction is over, we compute a second segmentation of the scene into hypothesized objects. Now, a verified hypothesis must meet two conditions: First, it is found in the segmentation before the interaction and is reliably matched with a facet after the interaction. And second, the respective facet must have moved due to the interaction. If both conditions are met, we consider the hypothesis to be verified. Note that a single interaction in clutter typically disturbs several objects, resuliting in several verified hypotheses. Even if only a single object is distrubed, it may be segmented into multiple (verified) facets.

A. *Computing Facets Similarity*

Given two facets from before and after the interaction, how can we determine whether they correspond to the same object facet? A naive answer is tracking the facet throughout the interaction. This would be computationally efficient and takes advantages of locality. However, because the manipulator is likely to obstruct our view of the object during manipulation, and because other objects in the pile may create temporary or partial obstructions, tracking becomes fragile and unreliable. Instead, we follow the paradigm of object detection: for every facet before the interaction, we search the results of segmentation after the interaction for a good match.

Facet matching computes the similarity between two facets by considering a variety of features. In our current implementation we have 8 different features: *(1) Relative Size* compares the number of points in the point cloud associated with each facet. *(2) Relative Area* compares the area occupied by the two facets in the projected RGB image. *(3) Average Color* and *(4) Color Histogram* compare the average HSV color and the intersection of the color histograms of the two facets. Finally, *(5-8) SIFT Matching* extracts and matches SIFT key-points from one facet to another. It then computes a rigid body transformation that best explains the mapping between the matched SIFT features. The rigid body transformation is applied to the first facet. Finally,

we measure the overlap between the transformed facet and the second facet. We determine overlap by averaging the pairwise distance between points in the two point clouds. Note that there are actually two *SIFT Matching* features: one computes SIFT matching from the smaller facet to the larger one (5) and the second from the larger facet to the smaller facet (7). Additionally, we have two binary features that indicate whether a rigid body transform was determined (features 6 and 8). If we find too few SIFT matches or there is too much disagreement between the feature matching and a good rigid body transform cannot be computed, the binary feature is set to false. Otherwise it is set to true. All feature are normalized to the range $[0, 1]$.

Given the 8 features above, we now have to compute a similarity score for every pair of segments. Naturally, some features are more indicative than others. In order to assign the appropriate weight to the features, we labeled examples of 15 scenes, each in 5 different configurations. For each scene, we had 10 pairs of before and after segmentations (every two configurations of the same scene). In total, we acquired labels for about 15000 pairs of facet, where only about 5% were positive examples. We assumed that our problem is linear and convex, and applied a Stochastic Gradient Descent algorithm [3] to learn the appropriate weights. The learned weights are as follows: Relative Size (4.618), Relative Area (2.543), Average Color (2.847), Color Histogram (6.329), SIFT large to small (1.222), SIFT large to small valid (0.418), SIFT small to large (3.628), SIFT small to large valid (0.348). Misclassification rates on test data were on average 5%, and all of them were false positives, meaning that two segments are not matched although they should be. We virtually encountered no true negatives (declaring segments to match when they should not).

B. Facet Matching

Our algorithm computes facet similarity scores for every pair of facets. Oftentimes, it is sufficient to pick for every facets the most similar facets as its match. In case the similarity score is below some threshold (e.g. 50%), the match is discarded. However, when an object has multiple facets with similar appearance, there may be several reasonable matches. This can be further complicated when several similar objects are present in the scene. To identify the optimal pairing of facets, we create a graph with two sets of vertices. One set contains a vertex for every facet before the interaction and the other set contains a vertex for every facet detected after the interaction. We connect a pair of vertices (one from each set) by an edge if the similarity likelihood is higher than 50%. Then, to resolve the ambiguity created by multiple edges connected to the same vertex, we compute bipartite matching [6], with the goal of maximizing the sum of log-likelihood. Effectively, we are extracting a subset of the pairing that maximizes the overall matching likelihood.

Finally, we only consider matched segments that have moved as a result of the interaction. If a facet remained stationary, re-detecting it after the interaction does not increase our confidence in the segmentation. The resulting matched-and-moved segments are verified segmentation hypotheses. We can now consider them for grasping. Figure 5 demonstrates the performance of this segmentation hypotheses verification process with three cluttered scenes. Objects vary in the type of material (rigid, flexible, articulated), dimensions, configuration (position and orientation), colors, and texture. The amount of motion and the number of moving segments is different in each example. The results show that all moved facets were correctly detected and matched (corresponding facets in Fig. 5 are color-coded).

VI. Action Selection and Compliant Interaction

Our algorithm generates two types of interactions with the environment: poking and grasping. During poking, the robot selects a facet based on the current segmentation of the scene, and pushes it parallel to the support surface for 3cm. After poking, the algorithm computes a list of verified segmentation hypotheses (matched and moved facets). The robot then selects one of the verified facets, and grasps it. In this paper, whenever the robot has multiple candidate facets to push or grasp, it selects one at random. We consider this as baseline for future work in which we intend to have the robot learn from its own experiences the best next action.

Poking and grasping in unstructured environments is challenging because the robot has only partial and inaccurate knowledge of the environment. This leads to uncertainty in modeling and localization. To overcome this uncertainties, we rely on a library of compliant controllers which maintain proper contact with the environment during the robot's motion by responding to the detected contact forces. The robot motion is planned using CHOMP [1] to minimize contact with the environment.

Our compliant controllers are described in detail in [15]. These controllers require only minimal information to be instantiated: the center of gravity and principal axes of the target object. To estimate the COG, we average the 3D position of all points in the facet's point-cloud. To estimate the principal axis of a facet, we compute principal components analysis (PCA) on the corresponding point cloud. These estimations assume that the density of a facet is uniformly distributed and the entire facet is visible to the robot. In practice, both assumption are frequently violated. Yet, they provide a good enough estimate. Figure 4(right column) shows an example of detecting the center of mass and principal axes.

We devised two compliant motion primitives: compliant grasping and compliant poking (pulling/pushing) primitives. These primitives are velocity-based operational space controllers. They rely on force feedback acquired by a force-torque sensor mounted on the robot's wrist. During the interaction, the robot's fingers are coordinated and position-controlled. The hand's configuration for both primitives is instantiated from perception (COG and principal axes of a facet). In this paper, we use a single cup-like hand pre-shape (Fig. 6). The width of the pre-shape is determined by the length of the facet along the secondary axis. Our

Fig. 5: Verifying segmentation hypotheses: every row shows a cluttered scene before (first column) and after some interaction (third column), and the corresponding matched segments (color coded, second and fourth columns respectively). Matched segments correspond to the same object facet, and have moved due to the interaction. They are candidates for grasping. Matching works well for all types of facets: rigid, flexible, or part of an articulated object; for different colors, sizes, positions, and orientations; and for both small and large motion between the two views.

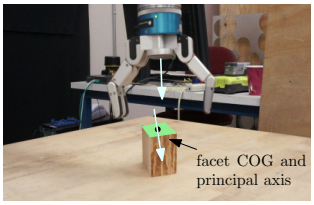


Fig. 6: The Barrett hand, assuming a cup-like pre-shape, is aligned with the top facet (in green) and located above the object

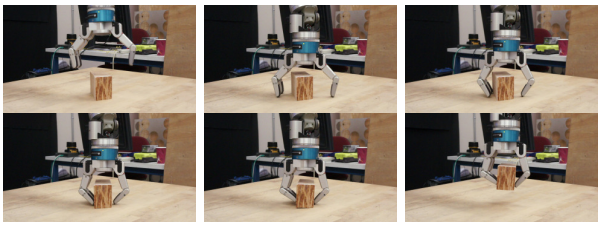


Fig. 7: The steps of compliant grasping: the Barrett hand, assumes a cup-like pre-shape, on top of the center of gravity and parallel to the principal axis. It moves compliantly towards the object until contact is detected. The finger close onto the object, and the object is grasped and transported to its destination.

current research considers others pre-shapes paramterized by the shape of the facet.

To grasp an object, we servo the hand along the palm's normal, until contact is detected between the fingertips and the support surface or the object. Then, we close the fingers, while the hand is simultaneously servo controlled (up or down) in compliance with the forces seen at the wrist in a closed-loop fashion. This ensures safe and proper contact between the fingertips and the support surface (see Fig.7). Note that our goal is to achieve a robust and firm grasp of an unknown object, and not positioning the fingertips at specific object locations.

Compliant poking is similar to the compliant grasping primitive. The launch pose of the hand is the same as for grasping, and the action itself is executed by seroving the hand towards (pull) or away from (push) the robot and in parallel to the support surface. We have thoroughly tested the implementation of the two compliant primitives on a robotic manipulator consisting of a 7-DOF Barrett Whole Arm Manipulator (WAM) and a 3-fingered Barrett hand. Experimental results and detailed discussion of the implementation is available in [15].

VII. Experimental Results

To evaluate our algorithm, we conducted dozens of experiments with a robotic manipulation system [1]. In our experiments, a variety of unknown (manmade and natural) objects were placed on a table in front of the robot (e.g., Fig. 1). Objects are placed in a pile. They often overlap and occlude each other to varying degrees. The robot's task is clearing the table by removing all objects into a box. The robot acquires RGB-D measurements of its environment using the Kinect.

Figures 8 and 9 show the steps taken by the robot to clear a pile of unknown objects in one of our experiments. The sequence begins with a set of objects placed next or on top of each other in an arbitrary configuration. The robot (1) segments the scene into facets, (2) pokes one facet

selected at random (using information extracted from the target facet to instantiate a compliant controller), (3) verifies its hypotheses by re-segmenting the scene and searching for matching moved facets, and (4) grasps one verified facet (again, using information extracted from the target facet to instantiate a compliant controller). The process continues until all objects have been removed.

In Fig. 8, the robot begins by poking the macaroni box. This action also disturbs the blocks and the shampoo. The robot now decides to grasp the bottle of shampoo. Next, the tissue box and the chunk of wood are pushed and grasped. The remaining two objects (macaroni box and toy blocks) are clustered closely together. The robot pokes the macaroni box, and then fails to grasp it because the hand hits the blocks. The failed grasp does disturb the blocks, so a second grasping attempt occurs (without poking). This time the robot successfully removes the blocks. Again, while removing the blocks the remaining item (macaroni box) is disturbed and no additional poking is necessary. The robot grasps the macaroni box and the process is completed successfully. Figure 9 shows the steps of the same experiment, as seen by the robot. The images are overlayed with the detected facets. Although in this paper we only allow one poke at a time, future work could consider multiple pokes before resegemnting the scene, leading to faster runtime.

In all our experiments the robot was able to remove all objects from the table and transport them into the box. In our approach, for n objects, the robot requires about $2n$ interactions: poking to verify segmentation and grasping for removing an object. Sometimes during grasping a neighbor object will be disturbed, allowing the robot to verify its segmentation hypothesis without an additional poke. And, occasionally poking does not verify any hypothesis, requiring additional interaction. In our experiments, the average was about 2 actions per object. This represents 3 times fewer interactions compared to the 6.6 actions per object in [5].

The execution time of the algorithm can be divided into three components: perception, poking and grasping. We measured the runtime for 100 instances of each. Segmenting a scene into facets takes on average 4 seconds. Poking an object requires an average of 10 seconds, and grasping and transporting the object take another 20 seconds. Thus, a typical sequence of segmenting-planning-poking-segmenting-verifying-planning-grasping-transporting -releasing requires about 34 seconds.

We have encountered four types of failure modes. First, perception may fail to segment an object if it is too small or incompatible with our sensor (e.g. depth cannot be measured for transparent objects). Second, poking an object can fail to move the object enough or can cause significant disturbance. In both cases facet matching may fail, and the robot will have to poke again. Third, grasping may fail because of collision (see for example Fig. 8(p)), if the object is too small or too large to fit in the hand, or if the object is slippery or flexible. We detect this failure using force sensing. Finally, occasionally an object will get out of the robot's reach or field of view. Future work could overcome these failures by considering better sensors, more dexterous hands, and allowing the robot to move about its environment.

VIII. Conclusion

We presented a fully integrated system for manipulating unknown objects in clutter. Our system incorporates sensing (RGB-D sensor), perception (segmentation and detection algorithms), control (a library of compliant controllers), and planning for collision avoidance. It enables a robot to extract 3D object segmentation hypotheses using an RGB-D sensor. Hypotheses are verified through deliberate interactions with the environment. Verified segmentation hypotheses are assumed to correspond to object facets. Our system relies on a library of compliant motion primitives, instantiated based on the extracted object facets, both for poking and grasping. Grasped objects are transported and released into a box.

Experimental results conducted with our manipulator (Fig. 1) demonstrate that our approach applies to a large variety of everyday objects placed in arbitrary configurations and with significant overlap. Our system continuously interacts with the environment until all objects placed in front of the robot are removed and placed in a target box. To the best of our knowledge, this is the first example of autonomous manipulation in clutter of unknown 3D objects.

We believe that this work is a prerequisite for more sophisticated pile manipulation. Our future work will rely on self-supervised learning to enable the robot to choose the best next action. For example, the right push may reveal much information, allowing the robot to proceed with a sequence of grasps. Or, in some cases, the robot may choose to rely on its initial hypothesis without verifying it (for example, if the segment is far from any other segment, or if the robot has seen this object in the past).

ACKNOWLEDGMENT

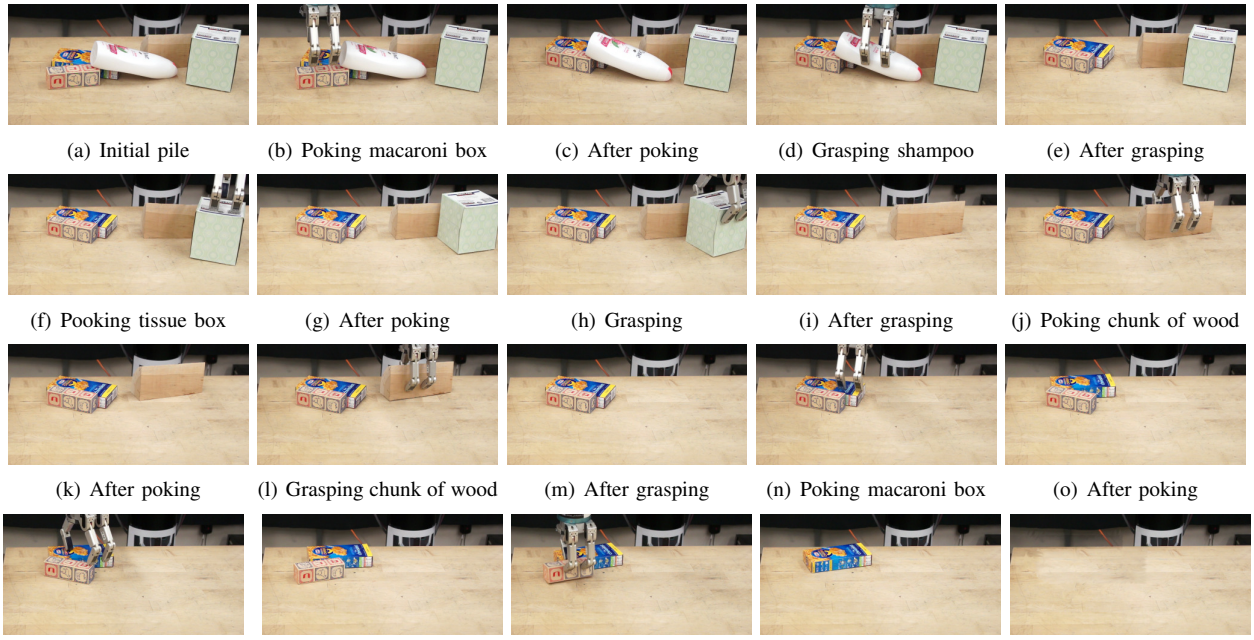
This work was conducted (in part) through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016. The authors also gratefully acknowledge alliance and a grant from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

References

[1] J. Bagnell, F. Cavalcanti, L. Cui, T. Galluzzo, M. Hebert, M. Kazemi, J. Libby, T. Y. Liu, N. S. Pollard, M. Pivtoraiko, J.-S. Valois, M. Klingensmith, and R. Zhu. An integrated system for autonomous robotics manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, may 2012.

[2] 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.

[3] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. NIPS Foundation (http://books.nips.cc), 2008.



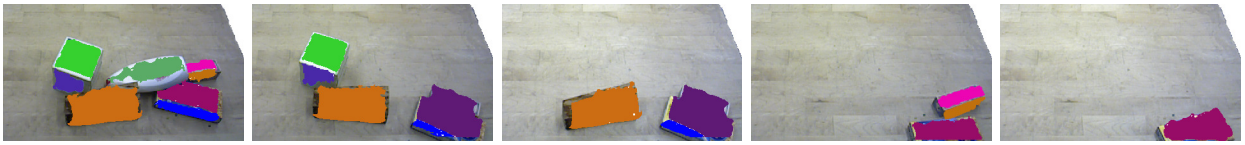
(a) Initial pile (b) Poking macaroni box (c) After poking (d) Grasping shampoo (e) After grasping

(f) Pooking tissue box (g) After poking (h) Grasping (i) After grasping (j) Poking chunk of wood

(k) After poking (l) Grasping chunk of wood (m) After grasping (n) Poking macaroni box (o) After poking

(p) Grasping macaroni box (failed) (q) After failed grasp (blocks disturbed) (r) Grasping blocks (s) After grasping (macaroni box disturbed) (t) After grasping macaroni box

Fig. 8: A sequence of interactions with a pile of unknown objects: a tissue box, a chunk of wood, a bottle of shampoo, a box of macaroni, and toy blocks. The algorithm switches between pushing to verify segmentation hypotheses and grasping to remove objects from the table. Here, 10 actions are required to remove 5 objects. Videos are available at http://www.dubikatz.com/autonomousManipulation.html



(a) Initial pile (b) Shampoo - removed (c) Tissue box - removed (d) Wood chunck - removed (e) Blocks - removed

Fig. 9: The robot's view of the scene during the experiment in Figure 8(p). Images are overlayed with the detected facets.

[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] L. Y. Chang, J. R. Smith, and D. Fox. Interactive singulation of objects from a pile. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3875–3882. IEEE, May 2012.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, September 2009.

[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] M. Gupta and G. S. Sukhatme. Using manipulation primitives for brick sorting in clutter. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3883–3889. IEEE, May 2012.

[10] A. Hauck, J. Ruttinger, M. Sorg, and G. Farber. Visual determination of 3D grasping points on unknown objects with a binocular camera system. In *IROS*, volume 1, pages 272–278, 1999.

[11] T. Hermans, J. M. Rehg, and A. Bobick. Guided pushing for object singulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012.

[12] D. Katz and O. Brock. Manipulating Articulated Objects with Interactive Perception. In *ICRA*, pages 272–277, Pasadena, CA, USA, May 2008. IEEE Press.

[13] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In *ISER*, India, 2010.

[14] D. Katz, Y. Pyuro, and O. Brock. Learning to Manipulate Articulated Objects in Unstructured Environments Using a Grounded Relational Representation. In *RSS*, pages 254–261, Zurich, Switzerland, June 2008.

[15] 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.

[16] 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.

[17] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[18] D. Marr. *Vision*. H. Freeman and Co., July 1982.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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.