Perceiving, learning, and exploiting object affordances for autonomous pile manipulation

Dov Katz, Arun Venkatraman, Moslem Kazemi, J. Andrew Bagnell & Anthony Stentz

Autonomous Robots

ISSN 0929-5593

Auton Robot
DOI 10.1007/s10514-014-9407-y

Auton Robot
DOI 10.1007/s10514-014-9407-y

Perceiving, learning, and exploiting object affordances for autonomous pile manipulation

Dov Katz · Arun Venkatraman · Moslem Kazemi · J. Andrew Bagnell · Anthony Stentz

Received: 26 September 2013 / Accepted: 10 July 2014

Abstract Autonomous manipulation in unstructured environments will enable a large variety of exciting and important applications. Despite its promise, autonomous manipulation remains largely unsolved. Even the most rudimentary manipulation task—such as removing objects from a pile—remains challenging for robots. We identify three major challenges that must be addressed to enable autonomous manipulation: object segmentation, action selection, and motion generation. These challenges become more pronounced when unknown man-made or natural objects are cluttered together in a pile. We present a system capable of manipulating unknown objects in such an environment. Our robot is tasked with clearing a table by removing objects from a pile and placing them into a bin. To that end, we address the three aforementioned challenges. Our robot perceives the environment with an RGB-D sensor, segmenting the pile into object hypotheses using non-parametric surface models. Our system then computes the affordances of each object, and selects the best affordance and its associated action to execute. Finally, our robot instantiates the proper compliant motion primitive to safely execute the desired action. For efficient and reliable action selection, we developed a framework for supervised learning of manipulation expertise. To verify the performance of our system, we conducted dozens of trials and report on several hours of experiments involving more than 1,500 interactions. The results show that our learning-based approach for pile manipulation outperforms a common sense heuristic as well as a random strategy, and is on par with human action selection.

Keywords Robotics · Manipulation · Learning · Object affordances

1 Introduction

Robots have been extremely successful for the past few decades in performing complex manipulation tasks on factory floors. However, they have had very limited success in our everyday environments. In our homes and offices, autonomous manipulation remains largely unsolved. Robots cannot perform simple everyday tasks such as clearing a pile of toys in the living room, tidying up a messy dinning table, or sorting a box of unused items in the garage. Why is it that robots can reliably perform pick-and-place tasks on the factory floor, yet miserably fail to do so anywhere else?

Let us consider a typical pick-and-place task of unknown objects in a pile. Manipulating such a pile requires close integration between perception, planning, and motion generation. The robot must move with care to avoid damage to itself and the environment, perform the task quickly and make as few assumptions as possible about the environment and the pile. On the factory floor, there exists near perfect control over the environment, removing the need for a robot to interact with clutter and piles of unknown objects. Therefore, many of the situations that occur in everyday-robotics and the resulting challenges in perception, decision making, and motion generation are eliminated or greatly reduced.

In this work, we identify several key prerequisites for manipulating a pile of unknown objects. First, the robot must acquire pertinent knowledge for interacting with individual objects in the pile. This is difficult because object segmentation remains an open problem, and is particularly challenging for a pile of overlapping and unknown objects. Because we cannot rely on prior object models, as the pile may contain

D. Katz (✉) · A. Venkatraman · M. Kazemi · J. A. Bagnell · A. Stentz
The Robotics Institute, Carnegie Mellon University,
Pittsburgh, PA 15213, USA
e-mail: dubikatz@gmail.com

Published online: 01 August 2014

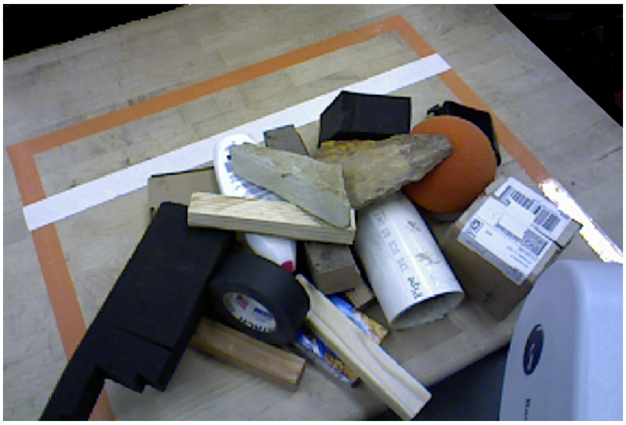


Fig. 1 Perceiving and manipulating unknown objects in a pile: Each detected object has a set of affordances (pushing, pulling or grasping). The robot selects the best next interaction for clearing the pile of unknown objects. The *orange boundaries* mark reachable space. The robot cannot grasp an object behind the *white boundary*, but may push or pull on it

natural objects, debris, and parts, the robot must hypothesize a segmentation of the environment into objects and compute for each object a set of affordances (Gibson 1977; Barck-Holst et al. 2009). We address object segmentation in a pile by proposing a geometry based segmentation algorithm.

Second, the robot must be able to choose which one of the affordances to execute next. Uninformed action selection can lead to slow performance, or worse, may damage the robot or objects in the pile. An intuitive heuristic—a set of rules—may be helpful in determining the next action, but is likely to fail often as it is difficult to anticipate the behavior of objects in a pile and the outcome of interactions. We propose a learning approach to manipulation. Our object representation exposes the structure of the pile and the affordances of the individual objects. Using this representation within a supervised learning framework, our robot is able to learn the necessary manipulation expertise to efficiently and reliably clear a pile of unknown objects (Fig. 1).

And finally, to avoid collision with other objects and enable careful interaction with target objects, the robot must generate safe motion trajectories. We developed a library of novel compliant controllers for poking, pulling and grasping unknown objects. These controllers are executed within a state-of-the-art motion planing pipeline.

In our experiments, the robot interacts with a pile of unknown objects placed on a table. The robot's task is to pick up individual objects and place them in a bin. We used both man-made and natural objects of varying shape, size, and appearance. Evaluating real-world manipulation performance is challenging. The interactions between the robot and objects and the interactions between objects in the pile are such that it is impossible to conduct the same experiment twice. Instead of repeating the exact same experience, we conducted numerous experiments using the same set of objects in arbitrary, different pile configurations. Thus, we conducted several hours of experiments consisting of over 1,500 interactions. Our results demonstrate that perceiving object affordances and learning to rank these affordances to determine the best next action facilitates a robust, efficient, and reliable pile clearing behavior. For transparency, we have uploaded many unedited videos of our experiments to http://www.youtube.com/user/pileRSS2013/videos.

The main contributions of this work are our novel geometric-based object segmentation method, a learning-based approach for pile manipulation, and a set of compliant control primitives which enable safe and reliable interaction with piles of unknown objects. This article is based on our prior work (Katz et al. 2013a,b; Kazemi et al. 2012; Bagnell et al. 2012).

2 Related work

In our work, we divide autonomous manipulation into three main components: perception, action selection, and motion generation. Perception segments the environment into objects and computes relevant features for manipulation and motion planning. Action selection determines object affordances, calculates the corresponding manipulation actions, and ranks the actions according to a learned metric predicting success. Finally, motion generation instantiates and executes the appropriate compliant controllers to achieve the selected action. Each of these three components relates to a vast body of work; the most relevant examples in each area are discussed below.

2.1 Object segmentation

Object segmentation is one of the fundamental problems in computer vision. It has been studied for decades, and yet extracting objects from a complex scene remains an open problem. Fortunately, to manipulate objects in a pile, it is unnecessary to have fully identify individual objects. Instead, we only need to capture properties of the environment that are specifically informative for grasping and manipulation.

Existing segmentation algorithms (David 2002; Zappella 2008) process an image and divide it into spatially contiguous regions sharing a particular visual 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. These assumptions are easily violated in a pile because of the significant overlap between objects. Thus, existing methods become brittle and unreliable. Moreover, color changes or texture gradients are not particularly informative for manipulation. Another proposed method has been to utilize visual edges (Hermans et al. 2012) for the task of object boundary detection. Image edges, however, do not reason about the 3-D structure of objects and may not provide useful information for manipulation.

An interesting category of segmentation algorithms leverage motion for identifying object boundaries. With these methods, the motion is either assumed to occur (Zhang et al. 2007; Stolkin et al. 2008; Goh and Vidal 2007) or it can be induced by the robot (Kenney et al. 2009; Katz and Brock 2008; van Hoof et al. 2013; Hausman et al. 2013). Although relative motion is a strong cue for segmentation, generating this motion in an unknown pile is oftentimes dangerous and undesirable. Our proposed method does not deliberately disturb the pile to generate motion for the purpose of segmentation. However, due to the nature of pile manipulation, relative motion occurs frequently, and when it does, our algorithm utilizes it to inform segmentation.

Segmentation can also be computed by considering 3-D geometry to determine the boundaries between objects (Taylor and Cowley 2011; Yang et al. 2010). Here, a boundary is defined as a depth discontinuity, and objects are modeled using parametric representations of predetermined shapes such as spheres, cylinders, and planes. These methods assume that objects can be described using a single basic shape. In practice, this is rarely the case. In fact, even if complex shapes are allowed, the geometry of the pile and the quick and abrupt way in which it can change make explicit shape modeling impractical. Nevertheless, for manipulating and grasping an object, its geometry can be informative. Therefore, our proposed method uses geometric properties to achieve segmentation. We use a non-parametric approach that considers both depth discontinuities as well as continuity in surface normal orientation to create object hypotheses.

Object segmentation in unstructured environments is unsolved. We believe that perception alone cannot be expected to provide models that are immediately actionable. This is because, without assuming prior knowledge, every segmentation algorithm becomes less reliable in clutter. The same applies to the method we propose. Therefore, we complement our segmentation algorithm with learning which enables the robot to identify unreliable segments and improve its action selection based on partial and unreliable perceptual information. This increases the performance of our system and can be considered as a means to improve the performance of perception with experience. We believe that such integration between all aspects of autonomous manipulation is essential for successful task execution in unstructured environments.

2.2 Learning manipulation expertise

For every object segmented by perception, our method instantiates a controller (or several controllers) to safely interact with the object. These potential interactions represent the object affordances (Barck-Holst et al. 2009; Gibson 1977). Choosing which of the possible actions to take is important: an action may be more or less likely to succeed, safe or dangerous, free up space around an object or condense the pile. The sequence of actions determines the number of interactions necessary to clear the pile. Thus, choosing the next best action is crucial for efficiency. Our method uses supervised learning to score and rank the objects' affordances.

Learning manipulation expertise is challenging because of the large state space associated with perceiving and manipulating objects. It is virtually impossible to encounter the same state twice. In practice, learning manipulation from real-world data is also challenging because perception may fail or provide unreliable answers. To combat this and the time-intensive nature of real-world collection, prior work in the field have used renderers to generate visual features for training data for grasping locations (Saxena et al. 2008) or simulate enough robot–object-interactions (Ugur et al. 2012) in order to automatically find a meaningful clustering of motion primitives to learn affordances. However, using real, collected data has the advantage that the training samples come from a similar distribution as at test time. In comparison with Ugur et al. (2012) and Saxena et al. (2008), we learn from real-sensor collected data. We also show that learning the affordances for object hypothesis from a simple set of pre-programmed motion primitives containing a pull, push, and two axis-aligned grasping actions is enough to achieve successful manipulation for clearing a pile. Interesting examples in the literature that apply learning to manipulation tasks include using relational reinforcement learning to learn a policy for modeling articulated objects (Katz et al. 2008) or for manipulating basic objects such as cubes, cylinders and spheres (Lang and Toussaint 2010). There has also been work to learn the effects of pushing objects from geometrically grounded features (Hermans et al. 2013). Additionally, supervised learning has been used to find and rank multi-contact grasp locations on objects in partially cluttered scenes (Le et al. 2010). However, learning grasping among other manipulation skills in densely cluttered unstructured environments, such as in piles of objects, remains largely unsolved.

Our work most closely resembles recent work on dense clutter and pile manipulation (Chang et al. 2012; Gupta and Sukhatme 2012). Chang et al. (2012) present a framework for object singulation with a final objective to grasp and remove relatively flat objects (e.g. candy bars) from the table. In this work, the framework chooses pushing actions until an object is spatially separated enough for the robot to safely grasp it. We address these issues by learning when grasps are likely to be successfully executed and we utilize force-feedback compliant motions to safely grasp objects. In our framework, for the similar task of object clearing, we can greedily exe-

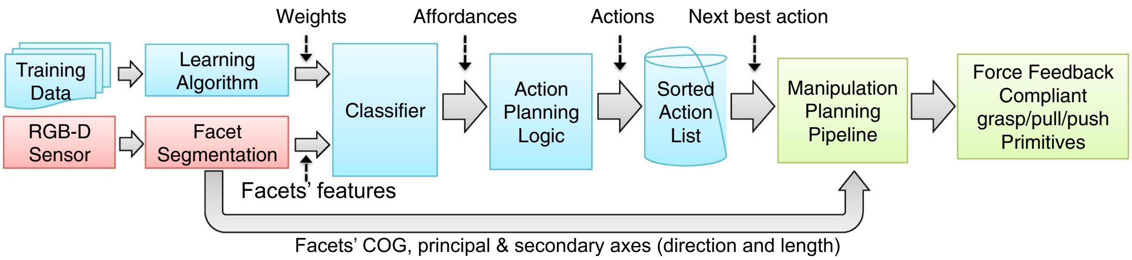


Fig. 2 System overview: perception (*red*) generates a segmentation of the scene into facets. Information about individual facets is used by learning (*blue*) to classify and score the affordances of each facet. Then, actions are ranked according to their scores, and the selected action is executed by instantiating a compliant controller (*green*)

cute any predicted grasping affordancess and use pushing and pulling motions when none are available. Gupta et al. (2012), considers simple motion primitives in order to singulate and remove Lego blocks from a table. Here, the robot and pipeline are built provided with *a priori* knowledge of the object type. For example, the assumption of uniformly colored Lego blocks allows for Euclidean color clustering. Knowledge of the bricks also allows the robot to safely interact with the pile without concern of breaking either the objects or the robot's manipulator. Our work extends (Chang et al. 2012; Gupta and Sukhatme 2012), augmenting the perception and manipulation portions of the pipeline to handle more complex objects (unknown, natural, and complex shapes), to consider more complex clutter (larger piles), and by introducing a new learning component to guide the interaction.

2.3 Motion generation

To execute a desired action, we first generate and execute a feasible trajectory to position the hand close to the target object. Then, we instantiate a compliant controller designed to achieve the desired manipulation behavior (pushing, pulling or grasping). We use CHOMP (Ratliff et al. 2009) to generate smooth trajectories and rely on a library of force feedback compliant motion primitives that are safe and appropriate for manipulation under uncertainty (Kazemi et al. 2012).

3 System overview

Our proposed system for manipulating unknown objects in a pile has three main components (Fig. 2): perception, learning-based action selection, and manipulation. Perception generates a set of object hypotheses, "facets" (Sect. 4). Action selection predicts the affordances of each object using trained SVM classifiers and chooses the best next action with the objective of clearing the pile safely and efficiently (Sect. 5). The manipulation pipeline then computes a motion plan and executes the appropriate compliant controller (Sect. 6).

4 Perceiving objects

Our perception pipeline is composed of two parts. The first computes a segmentation of the scene into facets (hypothesized object surfaces). For every facet, we extract the necessary information to instantiate our compliant controllers for pushing, pulling or grasping. The second part of the perception pipeline computes a set of visual features for each facet that is later used within a supervised learning framework to classify the affordances of each object.

4.1 Facet segmentation

To interact with unknown objects in a pile, we must first identify individual objects. Using 3-D information measured with an RGB-D camera, our algorithm segments the scene into hypothesized object facets. A facet is an approximately smooth circumscribed surface. An object facet is not necessarily a flat surface (plane), but rather a region maintaining continuity in both depth and the orientation of its surface normals. Dividing an object into facets is intuitive and repeatable under changes of perspective, lighting condition, and partial occlusion.

Facet detection is composed of the following three steps: computing depth discontinuities, estimating surface normals, and color-based 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 a device-specific threshold, the pixel is marked as a depth discontinuity. A 2cm threshold was used due to the resolution of our RGB-D sensor (Kinect). The surface normal at every point of the 3-D point cloud is estimated by fitting a local plane to the neighborhood of the point. We then compute the

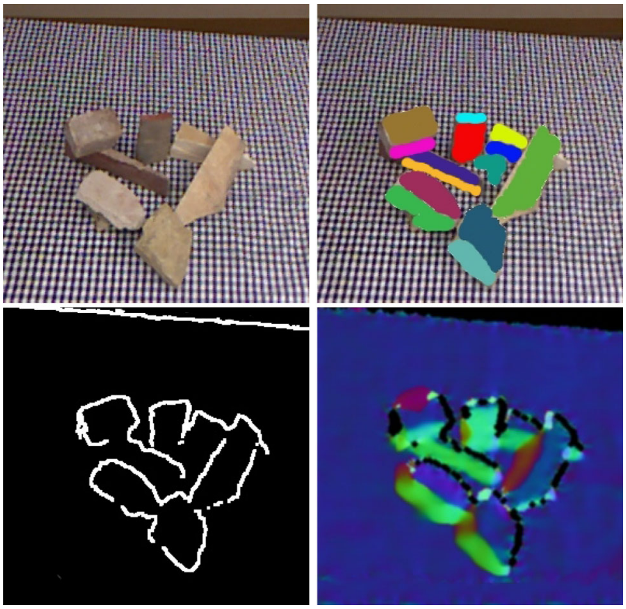


Fig. 3 Facet detection algorithm: The input (*top left*) is an RGB-D image. The algorithm extracts geometric discontinuities: depth discontinuities (*bottom left*) and normal discontinuities (*bottom right*). Finally, we merge depth and normals into a single RGB image. Object facets (*top right*) are extracted by computing color segmentation on that image

normal to that plane using least-square plane fitting. Figure 3 provides a visualization of the surface normals.

Next, we extract regions that are continuous in depth and surface normals. To exploit existing segmentation algorithms, we map the three Euler angles of every normal onto the three image color channels (RGB). We then overlay the depth discontinuities onto the color representation of the surface normals to form a color discontinuity where there is depth discontinuity. We have thus represented the facet segmentation problem as a standard color segmentation problem of extracting contiguous color regions (Comaniciu and Meer 2002). Therefore, we can extract facets using mean-shift segmentation, a standard color segmentation algorithm. More details and an experimental evaluation of facet detection is available in Katz et al. (2013a). Our contribution compared to that in Katz et al. (2013a) is algorithmic. Our version is more efficient and uses GPU acceleration where possible. This allows up to a x10 runtime speedup, which is essential for real-world manipulation.

Every segmented facet represents a hypothesized region where the robot can interact with the pile. For every facet, we compute its center of gravity (COG), the principal and secondary axes, and the length of each axis. We compute the COG of a facet by averaging the 3-D positions of the associated point cloud. We determine the principal and secondary axes by performing principal components analysis (PCA) on the region's corresponding 3-D point cloud. The length of each axis is the largest distance between a pair of points on or very close to the axis. Figure 4 illustrates the output of this process. We utilize this information to populate our robot's environment from which we can then instantiate our manipulation pipeline after action selection.

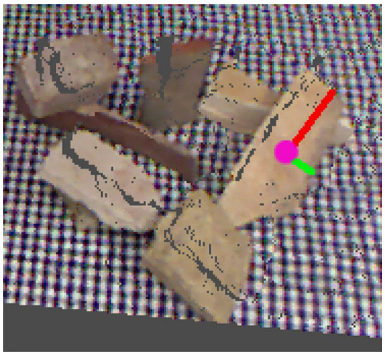


Fig. 4 Extracting information for manipulation: our algorithm computes the COG (*pink circle*), principal axis (*red*) and secondary axis (*green*) for every facet. This information together with the length of each axis suffices to instantiate our compliant controllers for pushing, pulling or grasping

Facet detection has two main limitations. First, our sensor (Kinect) cannot perceive reflective materials. And second, our method is unable to distinguish between two objects that are touching each other (having no depth discontinuity) and simulatenously have similar surface normals. This could be solved by considering color, texture, and experience. Because the robot disturbs the pile throughout its interactions, this case does not persist, and therefore has limited impact on our performance. To help alleviate errors in facet segmentation stemming from 3-D sensor noise, calibration errors, etc., our manipulation pipeline consists of compliant controllers that use force-control to compensate for partial and noisy perception. Thus, we are able to react to uncertainty in visual perception upon interaction with the environment.

4.2 Facet affordances

With the list of segmented facets and the information necessary to instantiate any of our controllers for every facet, we must now decide what is the next best action considering that not every action is feasible or desirable. For example, grasping a facet may not be possible because of other objects in the pile around the facet, or pulling an object that lies underneath other objects could result in an unwanted significant disturbance of the pile, risking both the robot and the objects. Thus, we must determine what are the affordances of each facet, accounting for its surroundings and create a ranking in order to determine what action to take next. These affordances depend not only on the facet itself, but also on its surroundings and the robot's capabilities.

Table 1 lists the 41 features we compute to determine the affordance of each facet. This list can be easily extended

Table 1 List of facet features associated with affordances

#	Feature	Description
1	CloudSize	Number of 3-D points associated with the facet
2	FacetArea	Projected 2-D area associated with the facet
3	Distance	Facet's Euclidean distance from the robot
4	Height	Facet's Euclidean distance from the support surface
5	Length	Distance between the farthest points along the principal axis
6	Width	Distance between the farthest points along the secondary axis
7	LW-ratio	Ratio between the length and width of the facet
8	SurfaceAngle	Angle between the facet and the support surface. The facet is represented as the surface defined by the principal and secondary axes
9	MoveMatch	Robot's confidence in the facet segmentation. This is computed by considering two consecutive frames. If a facet was disturbed and it can be retrieved in the second frame, the robot's confidence in its segmentation increases. For more details about matching facets across view see Katz et al. (2013a)
10–41	FreeSpace	Density of 3-D points around a facet determines the amount of free space around it. For efficiency, we only consider the area close to the extreme points of both the primary and secondary axes. Free space is represented by measuring the number of 3-D points in 8 small cylinders for each end of each axis. The cylinders are of radius 0.5cm, start at 2cm below the facet and end at 5cm above the facet. This feature is motivated by the notion that an empty or nearly empty cylinder indicates room for the fingers

These features are used within our supervised learning framework to select the best next action

to include additional features. In the next section, we detail utilizing these features within a supervised learning framework to determine the actual affordances of a facet: can it be pushed, pulled, and/or grasped along its principal or secondary axis.

5 Learning object affordances

We developed a supervised learning approach to manipulation for computing facet affordances. This is the most significant contribution of our work. Learning relies on the 41 features computed by perception (see Table 1). For training data, we labeled 37 scenes containing a total of 550 facets computed from a variety of different objects (Fig. 5). For each scene we used two image frames. We initially setup the scene (first frame), and in some cases disturbed the scene (second frame). Labeling was done for the second frame. The motion caused by disturbing the scene (if any) was used to compute feature #9 in Table 1. We developed a graphical user interface for displaying and labeling the segmented facets. For each facet, five binary labels were assigned by the user: actionable, push, pull, grasp-P and grasp-S (grasping along the principal or secondary axis). Note that the labels are not mutually exclusive and do not represent a preferred action. Instead, they respectively indicate whether a facet chould be interacted with, pushed, pulled, or grasped along either axis. This type of labelling by the user was used as a ground truth measure of both feasibility (can an action be at all performed) along with some implicit desirability (does it make sense for an action to be executed on the object). The specific distribution of the training labels can be seen in Table 2.

To classify the affordances of a facet, we use a simple linear classifier (linear SVMs Bishop 2006) with SVM-light Joachims 1999) on our 41 features. Each feature is normalized by its variance and thresholded outside of two standard deviations[1].

To compute the efficacy of our 41 features for learning affordances, we trained each classifier with 450 randomly selected instances, and tested on the remaining 100. The

[1] We scale each feature f_i using its mean $E(f_i)$ and variance $V(f_i)$. $f_i^{scaled} = (f_i - E(f_i))/\sqrt{Var(f_i)}$. If a scaled feature is more than two standard deviations away from the mean, we cap f_i^{scaled} at either -2 or 2. Finally, we divide f_i^{scaled} by 2 to guarantee that all features are in the range $[-1, 1]$.

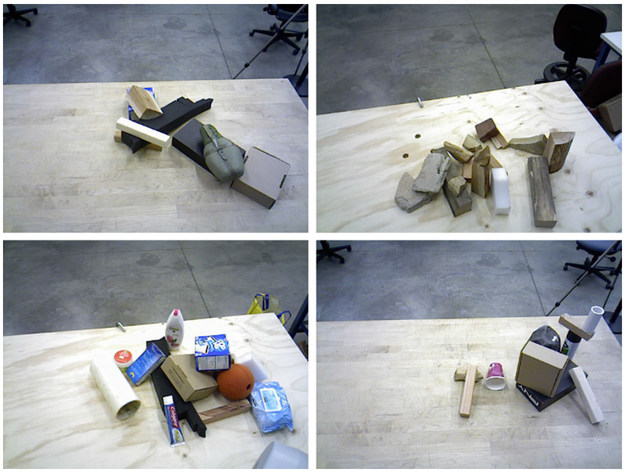


Fig. 5 Example scenes: training scenes were made by creating piles and clutter with numerous natural and man-made objects. To try to learn a well generalized model for each affordance, a variety of objects and pile densities were used in the training set

Table 2 Classifying facet affordances: we compare the distribution of positive and negative instances in the training examples to the classification rate achieved after training

Class	% Positive	% Negative	% Classification rate
Actionable	75	25	81.20
Push	43	57	91.75
Pull	59	41	93.45
Grasp-P	26	74	80.34
Grasp-S	37	63	80.10

The results show significant improvement of 24.8, 80.8, 84.0, 24.4, and 46.2 % in the misclassification rate respectively for each of the aforementioned classes compared to the naive approach of selecting the most probable label for each class

resulting classification rates and the distribution of positive and negative labels in the training set are summarized in Table 2. Grasping affordances are correctly classified in 80 % of the cases and pushing and pulling are correctly classified in over 90 % of the cases. We are also able to detect when a facet is invalid (not actionable) in 81 % of the cases. This is important for recovering from segmentation errors. A more careful analysis of the results shows that most of our misclassifications ($\geq$90 %) are true negatives, implying that the learner is conservative in deciding to act, which results in safer behavior.

Given a new scene, the robot is now ready to compute a segmentation, determine facet affordances, and rank the actions according to the score computed for every $< facet, action >$ pair by the classifiers. In our experiments, we create an action list by first adding the top 3 grasping actions followed by the top 3 pushing or pulling actions. Finally, we add all remaining actions (sorted by score). When an action cannot be performed (either because the planner detects a possible collision or because a trajectory to the goal configuration is infeasible), we continue to the next action in the list. Our bias towards grasping actions was motivated by the target objective of picking and placing objects into a bin. In future work, we intend to replace this action planning logic with reinforcement learning, allowing the robot to automatically learn the appropriate sorting of actions. This will enable us to develop simple strategies and learn from experience the appropriate scaling between the scores of the different classes.

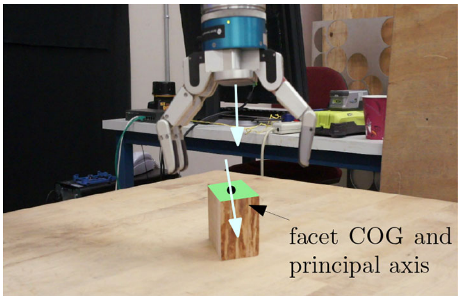


Fig. 6 To manipulate a facet, we first compute an action launch pose (i.e., robot hand/end-effector pose) based on the COG and orientation of the facet: the hand's palm is chosen parallel to the facet's principal and secondary axes, and its position is set with an offset above the facet's COG

6 Compliant motion primitives

To interact with the environment, we propose three types of parameterized controllers: pushing, pulling and grasping. Each controller is instantiated by perception based on the computed COG, principal and secondary axes, and the length of each axis. These controllers are inspired by and extend on the compliant grasping primitives developed in Kazemi et al. (2012). Interacting with unknown objects in a pile is challenging because the robot has only partial and inaccurate knowledge of the shape and configuration of objects. Thus, our controllers must be robust to uncertainty in modeling and localization. Inspired human grasping behaviors and as shown by Kazemi et al. (2012), the notion of compliance offers safe and robust manipulation in particular in cluttered environments.

To manipulate a facet, we first compute an action launch pose (i.e., robot hand/end-effector pose) based on the COG and orientation of the facet: the hand's palm is chosen parallel to the facet's principal and secondary axes, and its position is set with an offset above the facet's COG (see Fig. 6).

Given the kinematic model of the robot we perform inverse kinematics to obtain a feasible configuration for the robot to reach the desired hand/end-effector pose. Due to the kinematic redundancy of our system, for a given end-effector pose there may exist more than one feasible configuration to achieve the same end-effector pose. We perform an IK ranking to pick the "best" collision-free configuration according to a cost function based on the closeness to joint limits.

Our system uses CHOMP (Bagnell et al. 2012) to plan a smooth collision-free trajectory to the desired configuration. Then, we execute a compliant controller which maintains proper contact with the environment by responding to the detected contact forces. Our compliant controllers support pushing, pulling and grasping (either along the principal axis or the secondary axis). They are velocity-based operational space controllers, relying on force feedback acquired by a force/torque sensor mounted on the robot's wrist.

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 in compliance with the forces measured at the wrist. The fingers are coordinated using position-based controllers until they reach the object. This ensures safe and proper contact between the fingertips and the support surface. Figure 7 illustrates this process for grasping a block. Note that the palm is aligned with the facet and centered above the facet's COG. Also, the hand's aperture is determined by the length of the facet along the relevant axis.

Pushing and pulling begin in a similar way: we servo the hand along the palm's normal until contact is detected. To push an object, the hand is servo controlled along a vector parallel to the palm's normal and away from the robot until we either completed a trajectory of 5 cm, or the forces exerted onto the hand or fingers exceed a safety threshold. During the push motion the hand is also servo controlled along the palm's normal to maintain proper contact with the object. To pull an object, we apply force along the palm's normal (to maintain contact), while pulling the object towards the robot. Again, the action ends after moving for 5 cm or if an unsafe amount of force is detected. We have thoroughly tested the implementation of the three compliant controllers on a 7-DOF Barrett Whole Arm Manipulator (WAM) and a 3-fingered Barrett hand.

7 Experimental evaluation

To evaluate our system, we conducted dozens of experiments with a robotic manipulation system (Bagnell et al. 2012) developed for DARPA Autonomous Robotic Manipulation program. Videos of all of the experiments conducted for this paper are available at http://www.youtube.com/user/pileRSS2013/videos. In our experiments, a variety of unknown man-made and natural objects are placed in a pile on a table in front of the robot (e.g., Fig. 1). The objects overlap and occlude each other to varying degrees. The robot is composed of a 7-DOF WAM and a 3-finger Barrett hand equipped with force/torque sensor at the wrist. It acquires RGB-D measurements of the environment using a Kinect. The robot is tasked with clearing the table by removing all objects into a bin.

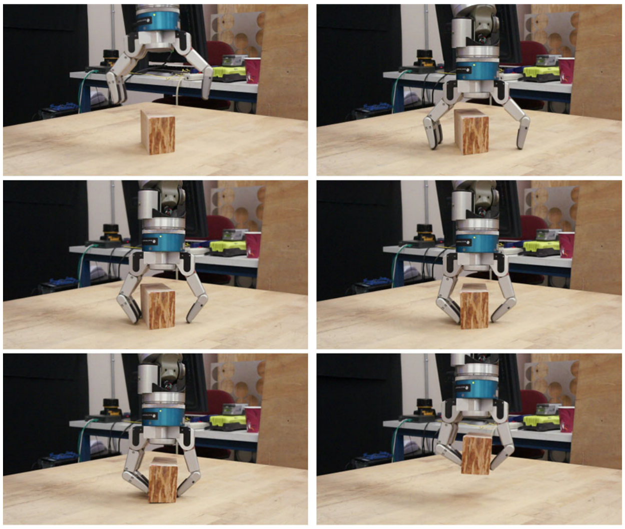


Fig. 7 The steps of compliant grasping: the Barrett hand assumes a cup-like pre-shape on top of the facet's center of gravity and is parallel to an axis of the facet. It moves towards the object until contact is detected. The fingers close onto the object while the hand is servo controlled in compliance with the forces due to contact with the support surface/object

We conducted three types of experiments. First, we evaluate the performance of 5 methods for selecting the next action: our learning-based approach, 2 random action selection strategies, a common-sense heuristic, and human-operator selected actions. Second, we analyze interesting instances highlighting the benefits of our learning-based approach. And finally, we compare the affordance classification of our learning method to action selection by human subjects.

7.1 Clearing piles of unknown objects

The main contribution of this work is developing a learning-based approach to manipulation. Our learned classifiers rank the affordances of segmented facets and generate a sorted list of actions. We compare the performance of learning to three other methods for action selection: random, heuristic-based selection, and a human operator. For random, we consider two strategies: select a facet at random and then either select one of our four action at random (all-random) or select only one of the two grasping actions at random (grasping-only-random). Our heuristic-based approach uses the following intuition:

1. Grasping the topmost object is safer and more likely to succeed
2. Grasping along the secondary (shorter) axis increases the chance of the object fitting into the robot's hand.
3. If an object is out of reach for grasping (behind the white line in Fig. 1), pulling is required.
4. Pushing can disturb/reorganize the pile and is therefore useful if the above actions cannot be performed.

We call this a common-sense heuristic as it encodes simple and seemingly obvious rules. It is possible to hard-code a more complicated heuristic utilizing all of the 41 features from Table 1; however, this can be difficult, time-consuming and brittle, in part due to errors such as noise and calibration offsets.

For the 'Human' experiments, a human operator selects the next action for the robot to execute using a graphical user interface to click on a facet and choose an action. In these experiments the human operator was one of the authors who is an expert in robotic grasping and is well familiar with the capabilities of the manipulation system.

Table 3 and Figs. 8 and 9 summarize the results of our experiments. We conducted extensive experiments consisting of 10 trials using each of our 5 methods for action selection. In our experiments, the robot attempted over 1,500 actions. In all experiments we used a randomly shuffled pile of the same 10 objects. When using all-random, the robot was never able to clear the pile. For example, the robot was able to remove only 2 objects after 50 actions. In Table 3 we present the results for the other 4 selection methods. We count the number of actions in every trial. A successful action occurs when the robot is able to plan a trajectory, executes it, and achieves the manipulation objective. A failed action occurs when the planned trajectory cannot be executed because of collision, the goal configuration cannot be reached by the robot, or the action itself fails (e.g. object slips out of hand). For each trial, we report the percentage of failed actions due to planning (% PF) and failure to achieve the manipulation goal (% EF).

Figure 8 shows the average number of actions and a standard deviation for each action selection strategy. The performance achieved by the (expert) human operator is not significantly better than our learning-based approach. In fact, lower performance was expected from the human operator if he/she was a non-expert with no background in robotic grasping and inexperienced with the manipulation capabilities of the system. Using our heuristic, the average number of actions is about 50 % higher than learning, and it increases by another 20 % when randomly selecting a grasping action. These results show the strength of our learning-based approach.

Figure 9 shows for each action selection strategy, the percentage of successful grasps out of the total attempted grasps. As expected, when a human selects a facet with a grasp action, the probability of success is the highest. Learning performs about 10 % worse. The likelihood of executing a successful grasp drops dramatically for the heuristic-based approach as well as for random. We believe that the results indicate that the human prefers preparatory actions (push/pull) to singulate objects over attempting difficult grasps. While this results in a higher grasping success rate, it also leads to more actions. Learning is more adventurous in choosing grasps. Although this results in more frequent failures to grasp, this strategy pays off as the overall number of actions needed is similar to what a person requires. Please see Table 3 which reports the detailed results for the action selection strategies.

7.2 Doing the right thing

Our second set of experiments analyzes interesting instances that demonstrate the behavior that was learned from the training data. In Fig. 10 (left), we presented the robot with a single large object (the detected facet marked in red). Learning classified this facet as negative for the "actionable" category, and did not attempt to interact with it. The other approaches (heuristic and random) kept interacting with the object without success.

The middle image in Fig. 10 contains three facets (red and green for the box that is out of the reachable area for grasping and blue for the ball). The top three actions ranked by learning are: pushing the orange ball (blue facet) into the reachable area, pulling the green facet and grasping the red facet along the longer axis. The heuristic would try to grasp the ball (difficult configuration, likely to fail) or pull the green facet (good). The red facet cannot be grasped (planning failure because grasping along the short axis would result in collision with the table), and since it cannot be grasped but yet is not outside the graspable zone, the heuristic will not try to pull it closer. Instead, it will keep pushing it towards the non-graspable zone. The right image in Fig. 10 shows cases where learning prefers pushing vs. pulling. As expected, learning classifies the green and red facets as positive for pushing and negative for pulling. The blue facet is classified as positive for pulling and negative for pushing.

In Fig. 11 we observe a frequent failure mode of our heuristic-based approach. Since it always grasps along the shorter axis and does not consider whether there is free space along this axis, it would randomly choose to grasp either the red or blue facets. The result strongly depends on the structure of the scene (left: success, right: failure).

In Fig. 12 we demonstrate that learning oftentimes generates sequences of interaction that benefit the robot. In this example, learning classifies both types of grasping as negative (the objects are too long for principal axis grasp and too close to each other to grasp along the shorter axis). Learning

Table 3 Results for 10 consecutive trials using the same 10 objects in arbitrary piles with our four action selection strategies: random-grasping-only, heuristic-based, learning-based, and a human operator-based

Pile	Actions					Failures		
	#	% GP	% GS	% PU	% PL	#	% EF	% PF
Random (grasping only)								
1	12	50	50	–	–	4	50	50
2	31	26	74	–	–	22	36	64
3	44	48	52	–	–	35	54	46
4	32	44	59	–	–	23	65	35
5	35	60	40	–	–	24	42	58
6	55	52	47	–	–	44	77	23
7	32	57	43	–	–	28	67	33
8	23	52	47	–	–	15	46	54
9	47	51	49	–	–	40	60	40
10	43	48	51	–	–	36	83	17
Heuristic								
1	35	–	94	3	3	25	100	0
2	10	–	100	0	0	0	0	0
3	22	–	90	5	5	10	80	20
4	63	–	92	5	3	51	16	84
5	33	–	100	0	0	23	43	57
6	14	–	100	0	0	4	100	0
7	65	–	87	5	8	52	13	87
8	19	–	95	0	5	12	34	66
9	17	–	88	0	12	8	37	63
10	23	–	83	4	13	10	50	50
Learning								
1	15	20	66	7	7	6	17	83
2	15	7	87	0	6	4	75	25
3	12	25	67	0	11	1	100	0
4	33	12	79	6	3	19	15	85
5	28	11	78	11	0	16	18	82
6	13	8	85	7	0	3	67	33
7	14	28	50	15	7	2	100	0
8	36	14	78	3	5	24	84	16
9	8	12	88	0	0	0	0	0
10	17	18	70	0	12	9	45	55
Human								
1	16	25	50	0	25	3	67	33
2	12	0	83	0	17	1	100	0
3	13	77	0	8	3	3	67	33
4	26	27	46	15	12	13	77	23
5	22	23	32	18	27	8	75	25
6	17	24	41	12	24	5	80	20
7	23	26	57	9	9	12	83	17
8	18	33	39	6	22	5	60	40
9	20	15	50	20	15	7	29	71
10	21	5	67	5	24	7	100	0

The columns are (left to right): trial id, number of actions to clear the pile, percentage of actions that were grasping-principal-axis (% GP), grasping-secondary-axis (% GS), pushing (% PU), and pulling actions (% PL), the total number of failures and the percentage of failures due to either execution (% EF) or planning (% PF)

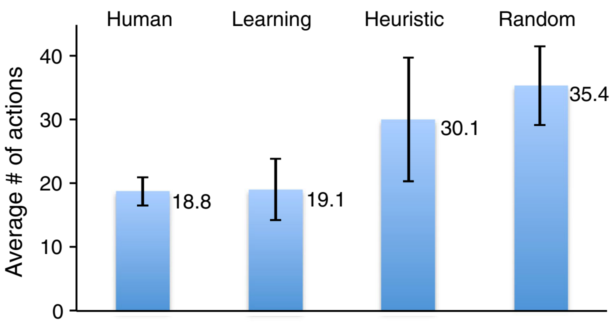


Fig. 8 The average number of actions required to remove all objects from the pile of 10 objects for all 4 action selection strategies. The results show that learning and human-operator action selection have similar performance, and are significantly better than the simpler methods (random and heuristic-based)

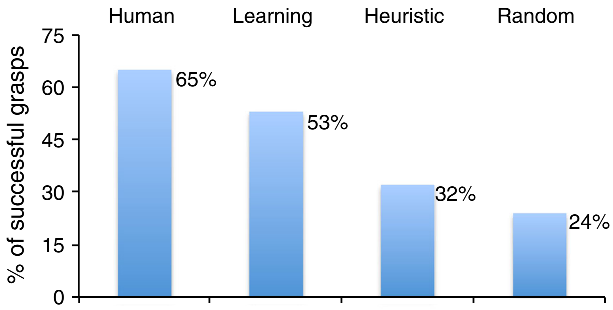


Fig. 9 The average fraction of successful grasps out of the total number of attempted grasps. The human action selection is more conservative, leading to higher success rate. Learning attempts more difficult grasps, which leads to more failures. However, note that both strategies require a similar number of action on average (Fig. 8). Heuristic-based and random action selection fail to execute a grasp in more than 60 and 70 % of the cases respectively

ranks pulling the red facet as the best next action, and after executing it (right image), grasping both facets becomes possible.

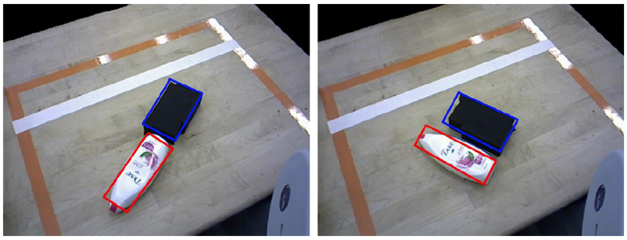


Fig. 11 Because our heuristic-based approach always grasps along the shorter axis and does not consider collision, the success depends on the structure of the scene. It would work if there is no collision along the secondary axis (*left*) and fail otherwise (*right*). Our learning-based approach identifies the difference and can choose between primary axis and secondary axis as necessary

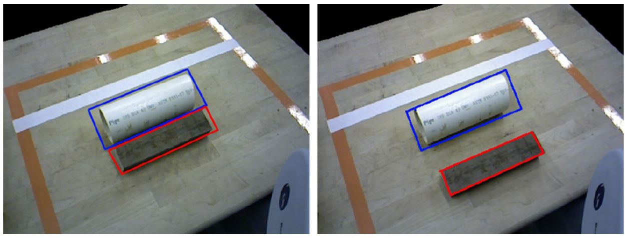


Fig. 12 Here (*left*) grasping along both axes will fail because the primary axis is too long to fit in the hand and using the secondary axis will result in collision. Learning anticipates this failure and prefers to pull the *red facet* first. In the next two steps, learning will remove the *red* and *blue facets* that are now separated and easy to grasp

7.3 Action selection: human versus learning

Figure 13 visualizes the ranking computed by our learning-based approach. For each affordance, the detected facets are color coded according to the output of the classifier: positive (green) and negative (red). For each affordance, the best facet is marked in bright green and the worst in bright red.

Interestingly, we informally asked 10 laypersons who were unfamiliar with our manipulation system to classify the facets into the 4 types of affordances. Qualitatively, we found that the classification suggested by the human subjects

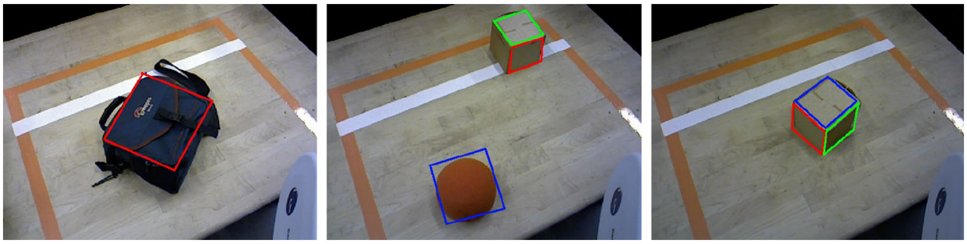


Fig. 10 Doing the right thing: analyzing the performance of our learning-based approach in interesting scenarios. *Left* Learning recognizes the object is too big for grasping; it decides not to interact with it. *Middle* Learning recognizes the *red facet* cannot be grasped (hand will collide with the table) and the green facet is outside the reachable zone grasping; it decides to pull the *box* (*green facet*) closer for grasping. The *ball* (*blue facet*) is too close for grasping; learning decides to push it towards the center. *Right* Learning correctly classifies the *blue facet* as good for pulling but not for pushing. Conversely, learning recommends pushing the *green* and *red facets* and not to pull on them

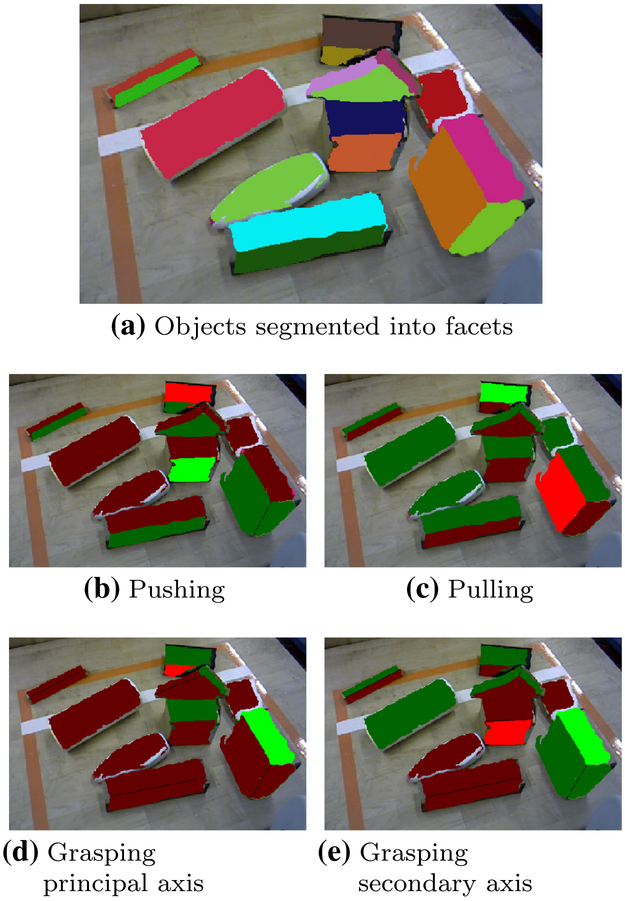

(a) Objects segmented into facets
(b) Pushing
(c) Pulling
(d) Grasping principal axis
(e) Grasping secondary axis

Fig. 13 An example scene composed of natural and man-made objects. We segmented the scene into facets and computed the classification assigned by learning for each affordance: positive (*green*) and negative (*red*). *Bright green* and *bright red* respectively represent the best and worst facet for each affordance. This classification was qualitatively similar to that suggested by 10 human subjects. Note that to simplify the task for the human subjects, this scene was constructed to be significantly simpler than those the robot was typically tasked with

was similar to that computed by our learning framework. We found these anecdotal observations very encouraging. However, we believe that analyzing the human action selection preferences deserves a thorough human subject study which is beyond the scope of this work. Only then one will be able to provide an informative qualitative and quantitative comparison with the learning-based action selection strategy proposed in this work.

8 Conclusion

This article describes our end-to-end system for autonomous manipulation in unstructured environments. We applied our system to the goal of removing unknown manmade and natural objects from a pile. In our approach, perception, learning and compliant motion generation are coupled together to achieve robust and safe task execution. We provided extensive experimental data demonstrating the merits of our approach.

Our robot relies on supervised learning to generate classifiers that relate unreliable partial perceptual information to action selection. Based on the learned classifiers, the robot interacts with the environment more efficiently than what was achieved with random interaction or by a common-sense heuristic. In comparison with a human-operator selecting the next best action for the robot to execute, our learning-based approach achieves, on average, the same number of actions necessary to clear the pile of objects.

Learning and generalizing manipulation knowledge enables the robot to autonomously interact with dense clutter. Learning becomes possible due to our novel algorithm for segmenting an unknown scene into hypothesized object facets. Perception provides the robot with a rich set of features. These features are informative for manipulation and grasping.

Our approach is not limited to supervised learning. While human-generated labels serve an excellent starting point, we expect our system to continue learning from its own experiences. In future work, our system will incorporate a reinforcement learning component enabling it to learn from its own decision making and real-world outcomes.

A key feature of our approach is safe interaction with the environment in the presence of uncertainty. We realize that perception is based on partial information due to occlusion. We also accept that action selection may make mistakes or select an action that is difficult to execute. Our system does not simply execute what action selection suggests. Instead, it utilizes a control strategy that guarantees safety and can recover from errors. Our controllers are compliant and rely on force sensing to determine how to execute a given task, as well as when to stop executing an action. Our compliant controllers overcome inevitable inaccuracies in perception and action selection while maintaining safe interaction with the environment.

We believe that there are many exciting extensions of this work with practical value. An immediate extension to our supervised learning approach is to use on-line self-supervised learning to adjust the learned weights of the classifiers based on the actual outcome of the robot's actions. We believe that this approach is essential for enabling autonomous manipulation in unstructured environments. Many additional perceptual cues could also be implemented to increase the range of what can be learned. And finally, our action selection is limited to a single step. It would be interesting to learn more complex strategies. This could significantly accelerate manipulation, in particular due to the complex nature of piles of unknown objects.

Acknowledgments 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 and also in part by Intel (Embedded Technology Intel Science and Technology Center). The authors also gratefully acknowledge funding under the DARPA Autonomous Robotic Manipulation Software Track (ARM-S) program.

References

Bagnell, J.A., Cavalcanti, F., Cui, L., Galluzzo, T., Hebert, M., Kazemi, M., Libby, J., Liu, T.Y., Pollard, N. S., Pivtoraiko, M., Valois, J-S., Klingensmith, M., & Zhu, R. (2012). An integrated system for autonomous robotics manipulation. In *IROS* (pp. 2955–2962).

Barck-Holst, C., Ralph, M., Holmar, F., & Kragic, D. (2009). Learning grasping affordance using probabilistic and ontological approaches. In *ICAR* (pp. 1–6).

Bishop, Christopher M., et al. (2006). *Pattern recognition and machine learning*. New York: Springer.

Chang, Lillian Y., Smith, Joshua R., & Fox, D., (2012). Interactive singulation of objects from a pile. In *ICRA*, (pp. 3875–3882).

Comaniciu, Dorin, & Meer, Peter. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Forsyth, D. A., & Ponce, J. (2002). *Computer Vision: A Modern Approach*. Englewood Cliffs: Prentice Hall Professional Technical Reference.

Gibson, J. J. (1977). *The theory of affordances, volume Perceiving* (pp. 67–82). Mahwah: Lawrence Erlbaum.

Goh, A., & Vidal, R., (2007). Segmenting motions of different types by unsupervised manifold clustering. In *CVPR* (pp. 1–6).

Gupta, M., & Sukhatme, G. (2012). Using manipulation primitives for brick sorting in clutter. In *ICRA* (pp. 3883–3889).

Hausman, K., Balint-Benczedi, F., Pangercic, D., Marton, Z.-C., Ueda, R., Okada, K., & Beetz, M. (2013). Tracking-based interactive segmentation of textureless objects. In *IEEE ICRA* (pp. 1122–1129).

Hermans, T., Rehg, J.M., & Bobick, A. (2012). Guided pushing for object singulation. In *IEEE IROS* (pp. 4783–4790).

Hermans, T., Li, F., Rehg, J.M., & Bobick, A.F. (2013). Learning contact locations for pushing and orienting unknown objects. In *IEEE-RAS International Conference on Humanoid Robotics*.

Joachims, T. (1999). Making large-scale svm learning practical. In B. Schlkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods—support vector learning*. Cambridge: MIT Press.

Katz, D., & Brock, O. (2008). Manipulating articulated objects with interactive perception. In *ICRA* (pp. 272–277).

Katz, D., Pyuro, Y., & Brock, O. (2008). Learning to manipulate articulated objects in unstructured environments using a grounded relational representation. In *RSS* (pp. 254–261), Zurich, Switzerland.

Katz, D., Kazemi, M., Bagnell, J.A., & Stentz, A. (2013a). Clearing a pile of unknown objects using interactive perception. In *ICRA* (pp. 154–161), Karlsruhe, Germany.

Katz, D., Venkatraman, A., Kazemi, M., Bagnell, J.A., & Stentz, A. (2013b). Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In *RSS*.

Kazemi, M., Valois, J-S., Bagnell, J.A., & Pollard, N. (2012) Robust object grasping using force compliant motion primitives. In *RSS*.

Kenney, J., Buckley, T., & Brock, O. (2009). Interactive segmentation for manipulation in unstructured environments. In *ICRA* (pp. 1343–48).

Lang, Tobias, & Toussaint, Marc. (2010). Planning with noisy probabilistic relational rules. *JAIR*, *39*, 1–49.

Le, Q.V., Kamm, D., Kara, A.F., Ng, A.Y. (2010). Learning to grasp objects with multiple contact points. In *ICRA* (pp. 5062–5069).

Ratliff, N., Zucker, M., Bagnell, J.A., & Srinivasa, S. (2009) CHOMP: Gradient optimization techniques for efficient motion planning. In *ICRA* (pp. 489–494).

Saxena, Ashutosh, Driemeyer, Justin, & Ng, Andrew Y. (2008). Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, *27*(2), 157–173.

Stolkin, R., Greig, A., Hodgetts, M., & Gilby, J. (2008). An EM/E-MRF Algorithm for adaptive model based tracking in extremely poor visibility. *Image and Vision Computing*, *26*(4), 480–495.

Taylor, C.J., & Cowley, A. (2011). In *RSS Workshop on RGB-D Cameras: Segmentation and analysis of rgb-d data*.

Ugur, E., Sahin, E., Oztop, E. (2012). Self-discovery of motor primitives and learning grasp affordances. In *IEEE IROS* (pp. 3260–3267).

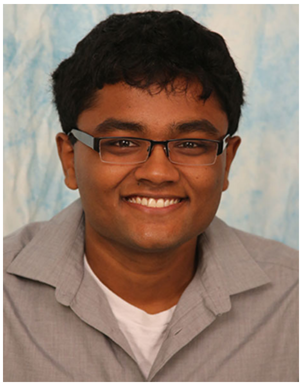
van Hoof, H., Kroemer, O., & Peters, J. (2013). Probabilistic interactive segmentation for anthropomorphic robots in cluttered environments. In *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*.

Yang, S.-W., Wang, C.-C., & Chang, C.-H. (2010). Ransac matching: Simultaneous registration and segmentation. In *ICRA* (pp. 1905–1912).

Zappella, L. (2008). Motion segmentation from feature trajectories. Master's thesis, University of Girona, Girona, Spain.

Zhang, J., Shi, F., Wang, J., Liu, Y. (2007). 3D motion segmentation from straight-line optical flow. In *Multimedia Content Analysis and Mining* (pp. 85–94). Springer, Berlin/Heidelberg.

Dov Katz is leading computer vision research for Oculus VR. He was a postdoctoral fellow with the Robotics Institute at Carnegie Mellon University. He received his Ph.D. and M.Sc. in Computer Science from the University of Massachusetts Amherst, and his B.Sc in Electrical Engineering and Computer Science from Tel-Aviv University in Israel. At Carnegie Mellon University, Katz was affiliated with the National Robotics Engineering Center (NREC). His research focuses on Machine Perception, Machine Learning, and Autonomous Manipulation.



Arun Venkatraman received a BS with honors in Electrical Engineering from the California Institute of Technology in 2012. He is currently a PhD student at Carnegie Mellon University in Pittsburgh, PA.

Moslem Kazemi is a Robotics Scientist at Brain Corporation in San Diego, California, where he is currently leading the development of robotic manipulation learning capabilities inspired by the functionality of human brain. He received his Ph.D. degree in Engineering Science from Simon Fraser University, Canada (2012), Masters degree in Industrial Systems Engineering from University of Regina, Canada (2004), and Bachelor degree in Computer Engineering from Sharif University of Technology, Iran (2000). In 2011 he joined the Carnegie Mellon University Robotics Institute as a Postdoctoral Fellow to work on the DARPA Autonomous Robotic Manipulation Software Track (ARM-S) project. In 2012, he was appointed as a Project Scientist at CMU and led the manipulation software development of the CMU ARM-S team. Dr. Kazemi's main research interests are: robotic manipulation and grasping, path planning and vision-based control of robotic arms, and smart systems software/hardware integration.

J. Andrew Bagnell is an Associate Professor in the Robotics Institute and Ma- chine Learning Departments at Carnegie Mellon University. He received his PhD from Carnegie Mellon in 2004. Bagnell's research focuses on the intersection of machine learning with computer vision, optimal control, and robotics. His interests in machine learning range from algorithmic and theoretical development to delivering fielded learning-based systems. Within robotics, Dr. Bagnell works on manipulation, 2- and 3-D computer vision, field and mobile robotics, and agile and dexterous control.

Anthony Stentz is a Research Professor at Carnegie Mellon University's Robotics Institute and Director of the National Robotics Engineering Center (NREC). Dr. Stentz has over 25 years experience in robotics and is a recognized expert in robot path planning, autonomous cross country navigation, and automating heavy mobile equipment used in industry.