Abstract—This paper proposes a new approach to localizing and grasping novel objects in clutter. The input to our algorithm is a point cloud and the geometric parameters of the robot hand. The algorithm locates hand configurations that are expected to result in good grasps. One of the challenges we address is that real-life point clouds are typically missing lots of data as a result of occlusions and object surface specularities. This can make perception hard. Our key innovation is to frame the grasp perception problem in terms of predicting the presence of antipodal grasps. The advantage of this approach is that an antipodal grasp is an objective and verifiable goal for localization that enables us to automate labeling of the training set. The result is larger training sets, less noisy labels, and better cross-validation that ultimately gives us better grasp prediction performance. Our experiments indicate that our approach will successfully grasp on average 96% of novel objects presented to the robot in a single-object scenario. In experiments where items are grasped from a disordered pile of 26 objects, our approach is able to grasp 90% of the objects successfully.

I. INTRODUCTION

Traditionally, robot grasping is understood in terms of two related subproblems: perception and planning. The goal of the perceptual component is to estimate the position and orientation (pose) of the object to be grasped. Then, grasp and motion planners are used to calculate where to move the robot arm and hand in order to grasp. While this approach can work in ideal scenarios, it has proven to be surprisingly difficult to localize the pose of objects in clutter precisely [8]. More recently, some researchers have proposed various methods of localizing grasps independently of object identity. Typically, these methods search an image, a range image, or a height map for local regions that are believed to be good grasps. There are two different variations on this approach. Some methods define a hand-coded local geometry that suggests the presence of a grasp [12, 9, 24]. Other methods use machine learning (typically a sliding window classifier) to locate good grasps. These methods require human-labeled training data in order to train the classifier [21, 10, 15, 6, 7]. Compared to the object-centric methods, grasp target localization methods have demonstrated the best performance grasping novel objects in clutter.

This paper proposes a new approach that improves upon the methods cited above in a couple of ways. First, we frame the problem of locating grasp targets in terms of localizing antipodal grasps. An antipodal grasp is one way of defining a “good” two-fingered grasp. Roughly speaking, it is a grasp formed when two fingers squeeze surfaces that have (approximately) equal and opposite surface normals [17]. The advantage of this approach is that an antipodal grasp is an objective and verifiable goal for localization. Compared to other approaches that rely on subjective human-labeled training data [21, 10, 15, 6, 7], this enables us to automate labeling of the training set. The result is larger training sets, less noisy labels, and better cross-validation that ultimately gives us better classification performance. In addition, in the future, this should make it possible to improve classification accuracy by incorporating multiple different sources of information. Another advantage of our approach is that we take a point cloud as input rather than a range image or a height map. This makes our algorithm easier to use in different sensing scenarios. For example, it is easy to use our algorithm to locate grasps in a voxelized map such as is generated by Kinect Fusion [13]. However, perhaps the most important advantage of our method is that it appears to work very well in practice. In particular, we can grasp approximately 96% of novel objects presented to the robot in isolation on the first try. This is a higher single-object grasp success rate than is reported by any of the other methods from the literature reviewed in Section I-B. Moreover, our algorithm can, on average, grasp and transport approximately 90% of the objects in a 26-object dense-clutter scenario such as shown in Figure 1.

A. Overview of the Approach

Our algorithm takes as input the geometric parameters of the robot hand. Perception happens in two basic steps. First, we generate a large number of grasp hypotheses by analyzing the point cloud relative to the hand geometry. Each grasp hypothesis describes the six-DOF pose of the robot hand and the finger configuration such that the fingers would sweep out some part of the environment if they were to close. Second, we classify each grasp hypothesis as either an antipodal grasp (i.e., a good grasp) or not. Once we have a set of hand poses that we believe to be antipodal grasps, we cluster the grasps, eliminate those that are kinematically infeasible, rank them using some kinematics-related heuristics, and execute a grasp
on the highest ranking candidate.

Perhaps the primary challenge faced by any approach that uses range sensor data to localize grasp targets is handling missing data correctly. In the ideal point cloud, there would be points on every exposed object surface. However, in practice, a point cloud created using data from one or more range sensors is nearly always missing significant numbers of points as a result of occlusions. This makes it hard to localize antipodal grasps. In order to handle the above “missing data problem”, we use classification to predict the presence of antipodal grasps. Our key idea is to create training data using multi-view registered point clouds. A multi-view registered point cloud is a cloud comprised of points generated by multiple range sensors where the points from different sensors have been aligned into the same reference frame. It turns out that it is sometimes possible to positively identify antipodal grasps in a multi-view point cloud by analyzing the surface normals and local geometry of various different surface patches in the cloud. Although we cannot locate all antipodal grasps this way, it is usually possible to localize some antipodal grasps. Furthermore, it is usually possible to identify some hand configurations that are definitely not antipodal grasps. Taken together, these positive and negative examples can be used to train a classifier to detect antipodal grasp affordances in the context of missing point cloud data. This turns out to be an effective way of localizing grasp targets. We get the best performance when using the resulting classifier to locate antipodal grasps in new multi-view point clouds. However, we still get good performance on single-view point clouds produced by a single range sensor.

B. Related Work

Perception has been one of the key challenges in practical robot grasping. Although we now have good approaches to grasp and motion planning [14], these algorithms require the ability to accurately localize objects in the environment. Recently, a new body of research approaches the problem of perception-for-grasping in a new way. Instead of attempting to localize objects and plan grasps, these approaches attempt to localize good grasp poses directly using perception. The idea of searching an image for grasp targets was probably explored first in Saxena’s early work that used a sliding window classifier to localize good grasps based on a broad collection of local visual features (and large amounts of human labeled training data) [21]. Later work extended this concept to range data [10] and explored a deep learning approach [15]. In [15], they report 84% grasp success rate using Baxter to grasp 30 novel objects (100 grasp trials) presented in a single-object setting (they obtain an 89% grasp success rate using the PR2). Fischinger and Vincze developed a method similar to the above [6,7]. Their work uses heightmaps instead of range images and develops a different Haar-like feature representation. In [7], they report a 92% single-object grasp success rate for 10 objects averaged over 50 grasp trials using the PR2. This work is particularly interesting because they demonstrate clutter results where the robot grasps and removes up to 10 piled objects from a box. They report that over six clear-the-box runs, their algorithm removes an average of 87% of the objects from the box.

Other approaches search a range image or point cloud for hand-coded geometries that are expected to be associated with a good grasp. For example Klingbeil et al. search a range image for a gripper-shaped pattern [12]. They report a 92% average grasp success rate for 120 single-object grasp trials over 6 novel objects using a PR2. Also, they report clutter experiments where the robot grasps and removes up to 7 objects from a pile. ten Pas and Platt propose an approach to localizing handles [24]. They report an 85% grasp success rate for 12 novel objects in a single-object scenario averaged over 144 grasp trials. They also report clutter experiments where they successfully grasp approximately 92% of the objects presented in a cluttered 5-object scenario with multiple attempts allowed.

A couple of approaches follow a template-based approach where grasps that are demonstrated on a set of training objects are generalized to new objects. For example, Herzog et al. learn to select a grasp template from a library based on features of the novel object [9]. They achieve an 87% success rate in single object experiments averaged over 95 grasp attempts for 38 different objects. Detry et al. grasp novel objects by modeling the geometry of local object shapes and fitting these shapes to new objects [4]. They report an 84% grasp success rate averaged over 55 grasp attempts with 9 objects using a Shunk hand. It also is worth mentioning another approach to grasping that is based on interacting with a stack of objects. For example, Katz et al. developed a different method of grasping novel objects based on interactively pushing the objects in order to improve object segmentation [11]. Chang et al. do something similar [2].

The approach presented in this paper can be compared directly with the above work. In this paper, we report a 96% success rate averaged over 115 grasps of 31 novel objects in a single-object setting. We also report clutter experiments where we remove 90% of the objects from a pile of 26 objects. These results are impressive for two reasons. First, a 96% success rate is higher than anything else reported above. Second, none of the work reported above considered a clutter scenario with as many as 26 objects.

II. LOCALIZING ANTIPODAL GRASPS

We frame the problem of localizing grasp targets in terms of locating antipodal grasps. An antipodal grasp is one way of defining a “good” two fingered grasp [17]. Two contacts are antipodal when their surface normals are opposite and intersect. In our context, we will require two surfaces of (approximately) antipodal points. Our approach has two basic steps. First, we generate a large number of grasp hypotheses by analyzing the point cloud relative to the hand geometry. Second, we classify each grasp hypothesis as either an antipodal grasp (i.e. a good grasp) or not.
We define a grasp hypothesis as the configuration of the hand and configuration of the fingers, such that: 1) the hand is not in collision with points in the cloud; 2) the hand axis is perpendicular to the page. Fig. 2. (a) Barrett hand forming a generalized two-fingered grasp. (b) Hand axis, \( \hat{a} \), is orthogonal to this plane (orthogonal to the page in Figure 2). Intuitively, this constraint makes sense because it “aligns” the hand with the object. Notice that this fixes two dimensions of orientation and reduces the sample space from six dimensions to four.

Algorithm 1 Calculate grasp hypotheses

\begin{algorithm}
\caption{Calculate grasp hypotheses}
\begin{algorithmic}[1]
\State \textbf{Input:} point cloud, \( \mathbb{C} \)
\State \textbf{Output:} antipodal grasp hypotheses, \( \mathbb{H} \)
\State 1: Preprocess \( \mathbb{C} \) (voxelize; workspace limits; etc.)
\For {i = 1 to n}
\State 2: sample \( s_i \in \mathbb{C} \) uniformly randomly
\State 3: \textbf{for} \( \alpha_i \) \textbf{do}
\State 4: estimate hand axis, \( \hat{a}_i \)
\State 5: calculate local neighborhood, \( N_d(s_i) \subseteq \mathbb{C} \)
\State 6: \( \Lambda(s_i) = \{ M_p : p \in N_d(s_i), \hat{a}_i^T(p - s_i) \leq 0.5h \} \)
\State 7: \( H_i = H_{yp}(\Lambda(s_i), \theta) \)
\State 8: \textbf{end for}
\State 9: \( \mathbb{H} = \bigcup_{i=1:n} H_i \)
\end{algorithmic}
\end{algorithm}

Algorithm 1 describes the process for generating grasp hypothesis samples in detail. First, we preprocess the point cloud, \( \mathbb{C} \subseteq \mathbb{R}^3 \), in the usual way by voxelizing and applying workspace limits (Step 1). Second, we uniformly randomly sample a set of \( n \) points, \( s_1, \ldots, s_n \in \mathbb{C} \), from the cloud (Step 2). We will search for grasp hypotheses in the vicinity of each sample. Next, for each sample, \( s_i \), we estimate the axis, \( \hat{a}_i \in \mathbb{R}^2 \), of minimum principal curvature using the method outlined in the Appendix (Step 3). The plane orthogonal to \( \hat{a}_i \) that passes through \( s_i \) will be called the cutting plane (shown in Figure 3(b)). The cutting plane is parametrized by a coordinate frame aligned with the surface normal and binormal. Next, for each of these samples, \( s_i \), we calculate a neighborhood of points, \( N_d(s_i) \in \mathbb{C} \), within a sphere centered at \( s_i \) with a radius equal to the outer diameter.

A. The class of robot hands for which our approach works

The approach developed in this paper can be used when the robot hand possesses a couple of important characteristics. First, we require the robot hand to be able to form a generalized two-fingered grasp. This is a grasp composed of two “virtual fingers”, where multiple fingers are allowed to move together in concert so as to function as a single “virtual finger” \[16\]. For example, the Barrett hand can form a generalized two-fingered grasp by moving two of the fingers together in opposition with the third (see Figure 2(a)); the second, we require the two fingers to move roughly in a plane orthogonal to the page shown in the figure. Intuitively, this constraint makes sense because it “aligns” the hand with the object. Notice that this fixes two dimensions of orientation and reduces the sample space from six dimensions to four.

B. Sampling Grasp Hypotheses

In order for the robot hand to grasp anything, the fingers must make contact with some part of the environment when they close. This motivates our definition of a grasp hypothesis. We define a grasp hypothesis to be a hand configuration (i.e. a pose of the hand and configuration of the fingers), such that: 1) the hand is not in collision with points in the cloud; 2) the hand closing region contains points in the cloud. Our approach to locating antipodal grasps will be to sample lots of grasp hypotheses and then to test each one to determine whether it is likely to be antipodal. Unfortunately, the space of hand poses that is to be sampled is six-dimensional: there are three dimensions of position and three of orientation. Generally, this is too many dimensions to sample densely.

A key idea of our approach is to reduce the dimension of the sampled space and to improve the expected quality of the grasp hypotheses by constraining the hand axis of each grasp hypothesis to be parallel to the axis of minimum principal curvature of the object surface in the vicinity of the grasp. For example, for the box in Figure 3(a,b), we would only consider grasp hypotheses where the hand axis was perpendicular to the plane shown in the figure. Intuitively, this constraint makes sense because it “aligns” the hand with the object. Notice that this fixes two dimensions of orientation and reduces the sample space from six dimensions to four.

Algorithm 1 describes the process for generating grasp hypothesis samples in detail. First, we preprocess the point cloud, \( \mathbb{C} \subseteq \mathbb{R}^3 \), in the usual way by voxelizing and applying workspace limits (Step 1). Second, we uniformly randomly sample a set of \( n \) points, \( s_1, \ldots, s_n \in \mathbb{C} \), from the cloud (Step 2). We will search for grasp hypotheses in the vicinity of each sample. Next, for each sample, \( s_i \), we estimate the axis, \( \hat{a}_i \in \mathbb{R}^2 \), of minimum principal curvature using the method outlined in the Appendix (Step 3). The plane orthogonal to \( \hat{a}_i \) that passes through \( s_i \) will be called the cutting plane (shown in Figure 3(b)). The cutting plane is parametrized by a coordinate frame aligned with the surface normal and binormal. Next, for each of these samples, \( s_i \), we calculate a neighborhood of points, \( N_d(s_i) \in \mathbb{C} \), within a sphere centered at \( s_i \) with a radius equal to the outer diameter.

1 The choice to sample randomly is arbitrary. Uniform grid sampling would work just as well.
2 The axis of minimum principal curvature and the axes of the cutting frame form a Darboux frame at \( s_i \).
that we are provided with a function, $H_{\text{empty}}$. This occurs in Step 7 of Algorithm 1. We assume $\Lambda(s_i)$ such that the 2d-hand is not in collision with any points in $\Lambda(s_i)$.

Now, the problem has become that of searching for antipodal grasp hypotheses within the local neighborhood, $N_d(s_i)$. This is a search over three dimensions of position and one dimension of orientation (rotation about the hand axis). We eliminate one dimension (thereby simplifying the search), by projecting the entire local search onto the cutting plane. Figure 3(c) and 4 show examples. Projecting the local search into the plane is a conservative step: we may miss some grasp hypotheses, but any hypotheses that we do find will satisfy the conditions for an antipodal grasp hypothesis. The projection happens in Step 6 of Algorithm 1. We crop points from $N_d(s_i)$ more than half the hand height from the cutting plane: that is, we crop all points $p \in N_d(s_i)$ such that $\hat{a}^T(p - s_i) > 0.5h$, where $h$ denotes the hand height. Then, the points are projected onto a basis for the cutting plane:

$$\Lambda(s_i) = \{Mp : p \in N_d(s_i); \hat{a}^T(p - s_i) \leq 0.5h\},$$

where $M \in \mathbb{R}^{2 \times 3}$ is a basis for cutting plane. Figure 3(c) illustrates $\Lambda(s_i)$ for the box in Figure 3(a,b).

Finally, we must generate a set of antipodal grasp hypotheses by finding 2d-hand positions and orientations in the plane such that the 2d-hand is not in collision with any points in $\Lambda(s_i)$ and such that the closing region of the hand is not empty. This occurs in Step 7 of Algorithm 1. We assume that we are provided with a function, $H_{\text{hyp}}(\Lambda(s_i), \theta)$, that locates antipodal grasp hypotheses in the plane for the planar neighborhood, $\Lambda(s_i)$, and the hand parameters, $\theta$. This is a three dimensional search (two dimensions of position and one of orientation), so brute force search is feasible. In the case of Baxter’s parallel jaw gripper, we were able to reduce this to a two dimensional search by making use of the geometry of the parallel jaws as follows. We iterate through a discretized set of hand orientations (in our case 8) and a discretized set of lateral hand positions (in our case 20). For each orientation and position, we move the hand forward until the hand is pushed out of the planar neighborhood or some part of the hand intersects a point in the planar cloud.

C. Creating The Training Set

Once we have a set of grasp hypotheses, the next task is to identify them as antipodal grasps or not. Ultimately, we will accomplish this using a support vector machine. However, in order to do that, we must obtain labeled data that can be used for training. The standard approach is to use human labeling $[21, 10, 15, 6, 7, 9]$. However, this is time consuming and it introduces noise into the training set. Instead, we extract training data from a set of one or more unlabeled multi-view registered point clouds.

Our key insight for obtaining training labels is to recognize that it is sometimes possible to determine whether a grasp hypothesis is actually an antipodal grasp by checking surface normals in the local point neighborhood. For example, we can verify that the grasp hypothesis on the right in Figure 4 is an antipodal grasp by noticing that the surface normals for the points closest to each finger are parallel with the closing direction of the fingers and outward pointing. Similarly, we can verify that the grasp hypothesis in the middle of Figure 4 is not an antipodal grasp by noticing that the surface normals are not orthogonal to the fingers. Notice that we do not have enough information to determine whether the grasp on the left of Figure 4 is antipodal, since the point cloud does not contain points on the opposite side of the box. When we cannot determine whether a grasp hypothesis is antipodal or not, we will refer to it as indeterminate.

Now that we have a way of determining whether a given grasp hypothesis is antipodal, not antipodal, or indeterminate, we can create the labeled training set. We require a set of point clouds from which the training information will be extracted. For each point cloud, we identify a set of antipodal grasp hypotheses using the method described in Section II-B. Then, we use the labeling method described above to classify each grasp hypothesis as either positive, negative, or indeterminate. Once labeling is complete, we discard the indeterminate samples and train the classifier using only the positive and negative examples.

It is important to mention that we require each point cloud from which training data will be obtained to be a multi-view registered point cloud. A multi-view registered point cloud is one that is created by taking range images from at least two different perspectives and aligning the corresponding point clouds using registration techniques. The reason for this requirement is intuitive: a single range sensor cannot perceive points on two opposite sides of an object. Without a multi-view point cloud, it is impossible to verify the existence of positive examples. Figure 5(a–c) illustrates indeterminate, negative, and positive grasps for the box example.
Most grasp hypotheses will be indeterminate in practical grasping scenarios. Even when it is possible to create a multi-view registered point cloud, occlusions caused by a cluttered environment will make it difficult to positively identify antipodal grasp affordances. Moreover, since it can be inconvenient to create a multi-view cloud, it is common to create a point cloud from a single range image. Therefore, we would like to have a way of predicting that a grasp hypothesis is antipodal or not in the presence of missing information. Our approach is to use classification.

In order to classify a grasp hypothesis, we need to encode it as some kind of feature descriptor. In the last few years, a variety of descriptors have been developed for use in point clouds including 3D SURF [13], FPFH [19], and SHOT [25]. The goal with these features is typically to describe a very small neighborhood uniquely in order to do registration. Our intent is different. We want to encode the geometry of the object surface contained in the closing region of the grasp hypothesis.

Our descriptor is inspired by the method that we used to sample grasp hypotheses. Recall that we projected local neighborhood points onto the cutting plane and that the coordinate frame of the cutting plane was aligned with the surface normal (Step 6 of Algorithm 1). Our approach is to convert this cutting plane projection into an image and apply standard 2D image classification techniques. In particular, we will convert the portion of the cutting plane that lies within the closing region of the hand into an image called grasp hypothesis image. There are a variety of potential classification methods that could be used. In this case, since we want something that will generalize appropriately across similarly shaped object surfaces, we want a method designed for object category recognition. We choose the HOG descriptor [3] because it is easy to use, but there are certainly lots of alternatives [26, 5]. Figure 6 illustrates the HOG representation corresponding to the given grasp hypothesis image. There are a variety of potential classification methods that could be used. In this case, since we want something that will generalize appropriately across similarly shaped object surfaces, we want a method designed for object category recognition. We choose the HOG descriptor [3] because it is easy to use, but there are certainly lots of alternatives [26, 5]. Figure 6 illustrates the HOG representation corresponding to the given grasp hypothesis image. We have found that our method is not terribly sensitive to the parameters of the HOG descriptor. In our implementation, we chose a cell size such that the grasp hypothesis image was covered by 10 × 12 cells with a standard 2 × 2 block size.

**D. Feature Representation**

Classification is performed using a support vector machine with a quadratic kernel. The training set is created from a set of registered point clouds in which we have identified positive and negative examples of antipodal grasps using the methods of Section II-C. For each of these positives and negatives, we create a HOG feature descriptor from the grasp hypothesis image built using the registered cloud (Section II-D). Recall that we throw out indeterminate grasp hypotheses from the training set because we cannot label them accurately just by checking local surface normals. However, since the goal of classification is to predict whether an indeterminate grasp hypothesis is actually an antipodal grasp or not, it seems that throwing out these examples would be a problem. We would like to have training data that is as similar as possible to the indeterminate grasps that we hope to classify. Therefore, we create indeterminate grasps “artificially” in the training set by throwing out data from one or more of the constituent views. For example, this is illustrated in Figure 7. Figure 7(a) shows the grasp hypothesis image produced using the original two-view registered cloud. If we throw out points perceived by the right camera and only keep the points produced by the left camera, we get the image in Figure 7(b). Similarly, if we throw out points perceived by the left camera, we get the image in Figure 7(c). We incorporate all three of these images into the training set. The images in Figures 7(b) and (c) show examples of what the system would see if it were using only one camera instead of two. This enables us to get examples of labeled indeterminate grasp hypotheses into the training set.

**E. Getting examples of indeterminate grasp hypotheses into the training set**

We characterize our ability to predict antipodal grasp affordances using cross validation. We obtained two-view registered point clouds for each of the 31 objects shown in Figure 8(b) presented individually from at least three different poses (121 two-view registered clouds in all). We generated training data from these point clouds using the methods already described in this this Section. The result was a total of 8409 training
images, of which 1791 were positive examples. We performed 10-fold cross validation and obtained an average of 97.49% of the samples correctly labeled. This suggests that we can expect to correctly predict whether a grasp hypothesis is antipodal or not 97.49% of the time.

G. Grasp Selection

Our approach to localizing antipodal grasps typically finds tens or hundreds of potential grasps, depending upon the number of objects in the scene. In this paper, we ignore object identity in this paper and allow the robot to grasp any object in the scene. We choose a grasp to attempt as follows. As a preprocessing step, we cluster potential grasps based on distance and orientation. Antipodal grasps that are nearby each other and that are roughly aligned in orientation are grouped together. Each cluster must be composed of a specified minimum number of constituent grasps. If a cluster is found, then we create a new antipodal grasp positioned at the mean of the cluster and oriented with the “average” orientation of the constituent grasps. The effect of this preprocessing step is to sparsify the grasp choices.

The next step is to select a grasp based on how easily it can be reached by the robot. First, we solve the inverse kinematics (IK) for each of the potential grasps and discard those for which no solution exists. The remaining grasps are ranked according to three prioritized criteria: 1) distance from joint limits; 2) distance from hand joint limits; 3) joint space distance traveled by the hand starting from a fixed pre-grasp arm configuration. Criterion #1 is important because it is sometimes the case that reaching accuracy degrades in the vicinity of the arm joint limits (this is true for Baxter, for example). Criterion #2 is important because it penalizes grasps that are very close to the hand minimum or maximum aperture and are therefore less robust to perceptual errors. Criterion #3 biases grasp selection toward grasps that will execute quickly. These three criteria are minimized in order of priority: first we select the set of grasps that minimize Criterion #1. Of those, we select those that minimize Criterion #2. Of those, we select those that minimize Criterion #3.

III. EXPERIMENTS

A. Hardware

Our robotic platform is Rethink’s Baxter Research Robot. We only use the right arm in our experiments. A two-finger parallel gripper is attached as the end-effector on this arm. To provide additional friction, we added rubber pads on the inner surface of the gripper fingers. The maximum amount by which the Baxter gripper can open and close is four centimeters. In these experiments, we adjusted the gripper so that the inner surfaces of the fingers were 7cm apart when the fingers were fully open and 3cm apart when the fingers were fully closed. We used KDL [22] as the inverse kinematics solver. We created two-view point clouds by registering together two point clouds produced by Asus Xtion Pro range sensors. The two Asus sensors were mounted approximately 60cm apart on a bar fixtured to the front of the robot (see Figure 1). We manually found an approximate relative transform between the two constituent clouds and adjusted that transform using Iterative Closest Point [1].

B. Single Object Experiments

In these experiments, we tested the ability of our system to localize, grasp, and lift novel objects placed in isolation on a tabletop. We trained the system using the set of five objects shown in Figure 8(a). Although the training object set is small, we chose training objects that we felt could be grasped in representative ways. For each training object, we obtained a two-view registered point cloud in each of four different orientations (three horizontal and one vertical) on a table. These point clouds were used for training using the
procedure outlined in Sections II-C, II-D, and II-E. We tested grasp performance with the 31 objects shown in Figure 8(b). These objects were chosen so that they could be grasped from any stable resting configuration by the Baxter gripper. In our gripper configuration, this means that at two sides of each object must have a section that is between 3cm and 7cm in diameter. We attempted to grasp each test object in three or four different orientations (depending upon object). A trial was considered a success only if the robot successfully localized, grasped, lifted, and transported the object to a box on the side of the robot where the object was dropped. For each test object/pose, we allowed the robot to make up to four grasp attempts without any human intervention.

The results are shown in Table I. The robot was able to successfully grasp, transport, and drop into a box objects on the first attempt approximately 96% of the time (5 failures out of 115 attempts). When allowing up to four attempts per object, the success rate increased to about 99% (one failure out of 115 attempts). Of the five first-attempt failures, one was caused by dropping the object after an initially successful grasp. The other four first-attempt failures were caused by perceptual errors: either the algorithm found the wrong grasp target or it did not find one at all. All four of these failures occurred for objects that are hard for the Asus Xtion Pro to see because of specularities. The one failure that persisted after four grasp attempts was for the Lysol bottle because it was push into a configuration from which the Asus was unable to see anything reasonable because of specularities.

C. Experiments in Dense Clutter

In these experiments, we characterize the ability of our approach to localize and grasp in dense clutter. We create a grasp scenario where 26 objects are are piled together in a shallow box (see Figure 9 and 10). The purpose of the box is to prevent objects from being knocked off the table. The box has small ramps on the sides to minimize grasp errors caused by the robot attempting to grasp the box itself. The 26 objects used in this experiment are a subset of the 31 objects used in the single object experiments. We eliminated the two squirt bottles and the bike helmet because they were too big to fit in the box. We eliminated the vacuum brush because it was specular. We also eliminated the heat gun because it was heavy and often slipped out of the grasp. At the beginning of each run, the 26 objects were placed in the box arbitrarily. We allowed up to 60 grasp attempts per run, i.e. the robot had 60 grasps with which to clear the 26 items from the box. A run was terminated prior to the 60 grasp limit when ten consecutive failures occurred or when five consecutive localization failures occurred. Typical failure modes included localization errors, collisions with the environment, or other types of control errors.

We ran the box-clearing experiment ten times in each of the following three different algorithm configurations:

1) No SVM classification. Grasp hypotheses identified by checking surface normals. Use a two-view registered point cloud.
2) SVM classification used to identify antipodal grasps. Use a two-view registered point cloud.
3) SVM classification used to identify antipodal grasps. Use a one-view registered point cloud.

All three algorithm configurations began by extracting grasp hypotheses as described in Section II-B. The first configuration evaluated how well our approach to labeling positive examples in the training performed as a grasp localization method in itself. No machine learning was used in this configuration. All we did was check whether the surface normals in the neighborhood of the grasp formed an antipodal grasp. The second configuration evaluated how well our full classification approach worked in a typical two-view point cloud. The third configuration evaluated our full method with a one-view point...
Fig. 11. One failure mode for our algorithm occurs when two objects are misperceived as a single object.

| Setting                  | % objects successfully removed from box | Average # of Trials |
|--------------------------|----------------------------------------|---------------------|
| Two-view, Antipodal      | 57.6%                                   | 41                  |
| Two-view, Classification | 90.3%                                   | 44.6                |
| One-view, Classification | 83.4%                                   | 50.5                |

This paper proposes a new approach to localizing and grasping novel objects. Perhaps our most important innovation is to frame the problem in terms of localizing antipodal grasps in point cloud data. The advantage of this approach is that an antipodal grasp is an objective and verifiable goal for localization. Compared to other approaches that rely on subjective human-labeled training data [21] [10] [15] [6] [7], this enables us to automate labeling of the training set. The result is larger training sets, less noisy labels, and better cross-validation that ultimately gives us better classification performance.

We evaluate our method by performing robot experiments. We achieve a 97.5% accuracy in cross-validation studies where the goal is to predict whether a grasp hypothesis is an antipodal grasp or not. In single-object robot experiments where the goal is to grasp novel objects presented to the robot in isolation, we achieve a 96% success rate on the first grasp attempt. In a dense clutter scenario comprised of 26 novel objects arranged in a disorganized pile, we are able to grasp, lift, and transport approximately 90% of the objects in a given setting. To our knowledge, all of the success rates described above are competitive or better than what other methods in the literature have been able to achieve.

APPENDIX

Perhaps the simplest way to calculate the axis of minimum principal curvature is to use surface normals as follows. First, estimate the surface normal, \( \mathbf{n}(p) \in S^2 \), at each point, \( p \in \mathbb{C} \), in the point cloud using standard techniques [20]. Let \( q \in \mathbb{C} \) be the point at which we want to estimate the axis of curvature. Let \( N_q \subseteq \mathbb{C} \) denote a neighborhood of points within distance \( r \) of \( q \): \( N_q = \{ p \in \mathbb{C} \cap B_r(q) \} \), where \( B_r(q) \) denotes the \( r \)-ball about point \( q \). Calculate the matrix \( M = \sum_{p \in N_q} \mathbf{n}(p)\mathbf{n}(p)^T \). The minimum principal curvature axis is in the direction of the Eigenvector with the smallest corresponding Eigenvalue.

One problem with the above approach is that it requires calculating surface normals everywhere in the cloud. This can be time consuming and imprecise. Instead, we do the following. First, calculate the point-neighborhood within the \( r \)-ball about \( q \): \( N_q = \{ p \in \mathbb{C} \cap B_r(q) \} \). Second, fit a quadric using Taubin’s method [23] [24] (this involves inverting a \( 10 \times 10 \) matrix). Third, sample surface normals from the quadric, and proceed as before: calculate \( M = \sum_{p \in N_q} \mathbf{n}(p)\mathbf{n}(p)^T \) and take the Eigenvector with the minimum corresponding Eigenvalue.

REFERENCES

[1] P. Besl and N. McKay. A method for registration of 3d shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence, 14(2):239256, 1992.
[2] Lillian Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 3875–3882. IEEE, 2012.
[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.
[4] Renaud Detry, Carl Henrik Ek, Marianna Madry, and Danica Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In IEEE International Conference on Robotics and Automation, 2013.

[5] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9):1627–1645, 2010.

[6] D. Fischinger and M. Vincze. Empty the basket - a shape based learning approach for grasping piles of unknown objects. In IEEE Int’l Conf. on Intelligent Robot Systems, 2012.

[7] David Fischinger, Markus Vincze, and Yun Jiang. Learning grasps for unknown objects in cluttered scenes. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 609–616. IEEE, 2013.

[8] J. Glover and S. Popovic. Bingham procrustean alignment for object detection in clutter. In IEEE Int’l Conf. on Intelligent Robot Systems, 2013.

[9] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In IEEE Int’l Conf. on Robotics and Automation, 2012.

[10] Y. Jiang, S. Moseson, and A. Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In IEEE Int’l Conference on Robotics and Automation, 2011.

[11] D. Katz, M. Kazemi, D. Bagnell, and A. Stentz. Clearing a pile of unknown objects using interactive perception. In IEEE Int’l Conf. on Robotics and Automation, 2013.

[12] E. Klingbeil, D. Rao, B. Carpenter, B. Ganapathi, A. Ng, and O. Khatib. Grasping with application to an autonomous checkout robot. In IEEE Int’l Conf. on Robotics and Automation, 2011.

[13] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough transform and 3d surf for robust three dimensional classification. In Computer Vision–ECCV 2010, pages 589–602. Springer, 2010.

[14] S. Lavalle. Planning Algorithms. Cambridge University Press, 2006.

[15] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. In Robotics: Science and Systems, 2013.

[16] C. MacKenzie and T. Iberall. The Grasping Hand. North-Holland, 1994.

[17] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. A mathematical introduction to robotic manipulation. CRC press, 1994.

[18] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Int’l Symposium on mixed and augmented reality, 2011.

[19] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (pfh) for 3d registration. In IEEE Int’l Conf. on Robots and Automation, 2009.

[20] R. Rusu, Z. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems, 56:927–941, 2008.

[21] A. Saxena, J. Driemeyer, and A. Ng. Robotic grasping of novel objects using vision. International Journal of Robotics Research, 27(4):157, 2008.

[22] R. Smits. KDL: Kinematics and Dynamics Library. http://www.orocos.org/kdl.

[23] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. IEEE Trans. PAMI, 13:1115–1138, November 1991.

[24] A. ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In Int’l Symposium on Experimental Robotics, 2014.

[25] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In Computer Vision–ECCV 2010, pages 356–369. Springer, 2010.

[26] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511. IEEE, 2001.