# Using Geometry to Detect Grasp Poses in 3D Point Clouds

Andreas ten Pas and Robert Platt

College of Computer and Information Science, Northeastern University
Boston, Massachusetts, USA

**Abstract.** This paper proposes a new approach to using machine learning to detect grasp poses on novel objects presented in clutter. The input to our algorithm is a point cloud and the geometric parameters of the robot hand. The output is a set of hand poses that are expected to be good grasps. There are two main contributions. First, we identify a set of necessary conditions on the geometry of a grasp that can be used to generate a set of grasp hypotheses. This helps focus grasp detection away from regions where no grasp can exist. Second, we show how geometric grasp conditions can be used to generate labeled datasets for the purpose of training the machine learning algorithm. This enables us to generate large amounts of training data and it grounds our training labels in grasp mechanics. Overall, our method achieves an average grasp success rate of 88% when grasping novels objects presented in isolation and an average success rate of 73% when grasping novel objects presented in dense clutter. This system is available as a ROS package at `http://wiki.ros.org/agile_grasp`.

## 1 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 an object to be grasped. Then, grasp and motion planners are used to calculate where to move the robot arm and hand in order to perform grasp. While this approach can work in ideal scenarios, it has proven to be surprisingly difficult to localize the pose of novel objects in clutter accurately [6]. More recently, researchers have proposed various grasp point detection methods that localize grasps independently of object identity. One class of approaches use a sliding window to detect regions of an RGBD image or a height map where a grasp is likely to succeed [16,8,4,5,12,10]. Other approaches extrapolate local "grasp prototypes" based on human-provided grasp demonstrations [3,7,11].

A missing element in the above works is that they do not leverage the geometry of grasping to improve detection. Grasp geometry has been studied extensively in the literature (for example [13,17]). Moreover, point clouds created using depth sensors would seem to be well suited for geometric reasoning.

In this paper, we propose an algorithm that detects grasps in a point cloud by predicting the presence of necessary and sufficient geometric conditions for grasping. The algorithm has two steps. First, we sample a large set of grasp hypotheses. Then, we classify those hypotheses as grasps or not using machine learning.

**Fig. 1.** Our algorithm is able to localize and grasp novel objects in dense clutter.

Geometric information is used in both steps. First, we use geometry to reduce the size of the sample space. A trivial necessary condition for a grasp to exist is that the hand must be collision-free and part of the object surface must be contained between the two fingers. We propose a sampling method that only produces hypotheses that satisfy this condition. This simple step should boost detection accuracy relative to approaches that consider every possible hand placement a valid hypothesis. The second way that our algorithm uses geometric information is to automatically label the training set. A necessary and sufficient condition for a two-finger grasp is an antipodal contact configuration (see Definition 1). Unfortunately, we cannot reliably detect an antipodal configuration in most real-world point clouds because of occlusions. However, it is nevertheless possible *sometimes* to verify a grasp using this condition. We use the antipodal condition to label a subset of grasp hypotheses in arbitrary point clouds containing ordinary graspable objects. We generate large amounts of training data this way because it is relatively easy to take lots of range images of ordinary objects. This is a huge advantage relative to approaches that depend on human annotations because large amounts of training data can significantly improve classification performance.

Our experiments indicate that the approach described above performs well in practice. We find that without using any machine learning and just using our collision-free sampling algorithm as a grasp detection method, we achieve a 73% grasp success rate for novel objects. This is remarkable because this is a trivially simple detection criterion. When a classification step is added to the process, our grasp success rate jumps to 88%. This success rate is competitive with the best results that have been reported. However, what is particularly interesting is the fact that our algorithm achieves an average 73% grasp success rate in dense clutter such as that shown in Figure 1. This is exciting because dense clutter is a worst-case scenario for grasping. Clutter creates lots of occlusions that make perception more difficult and obstacles that make reaching and grasping harder.

### 1.1 Related Work

The idea of searching an image for grasp targets independently of object identity 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 [16]. Later work extended this concept to range data [8] and explored a

deep learning approach [12]. In [12], they obtain an 84% success rate on Baxter and a 92% success rate on the PR2 for objects presented in isolation (averaged over 100 trials). Fischinger and Vincze developed a similar method that uses heightmaps instead of range images and develops a different Haar-like feature representation [4,5]. In [5], they report a 92% single-object grasp success rate 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 [10]. In our prior work, we developed an approach to localizing handles by searching a point cloud for a cylindrical shell [19]. Other 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 [7]. Detry *et. al* grasp novel objects by modeling the geometry of local object shapes and fitting these shapes to new objects [3]. Kroemer *et. al* propose an object affordance learning strategy where the system learns to match shape templates against various actions afforded by those templates [11]. Another class of approaches worth mentioning are based on interacting with a stack of objects. For example, Katz *et. al* developed a method of grasping novel objects based on interactively pushing the objects in order to improve object segmentation [9]. Chang *et al.* developed a method of segmenting objects by physically manipulating them [2]. The approach presented in this paper is distinguished from the above primarily because of the way we use geometric information. Our use of geometry to generate grasp hypotheses is novel. Moreover, our ability to generate large amounts of labeled training data could be very important for improving detection accuracy in the future. However, what is perhaps most important is that we demonstrate "reasonable" (73%) grasp success rates in dense clutter – arguably a worst-case scenario for grasping.

## 2 Approach

We frame the problem of localizing grasp targets in terms of locating *antipodal hands*, an idea that we introduce based on the concept of an antipodal grasp. In an antipodal grasp, the robot hand is able to apply opposite and co-linear forces at two points:

**Definition 1 (Nguyen [14]).** *A pair of point contacts with friction is* **antipodal** *if and only if the line connecting the contact points lies inside both friction cones* [1].

[1] A friction cone describes the space of normal and frictional forces that a point contact with friction can apply to the contacted surface [13].

If an antipodal grasp exists, then the robot can hold the object by applying sufficiently large forces along the line connecting the two contact points. In this paper, we restrict consideration to parallel jaw grippers – hands with parallel fingers and a single closing degree of freedom. Since a parallel jaw gripper can only apply forces along the (single) direction of gripper motion, we will additionally require the two contact points to lie along a line parallel to the direction of finger motion. Rather than localizing antipodal contact configurations directly, we will localize hand configurations where we expect an antipodal grasp to be achieved in the future when the hand closes. Let $\mathcal{W} \subseteq \mathbb{R}^3$ denote the robot workspace and let $\mathcal{O} \subseteq \mathcal{W}$ denote space occupied by objects or obstacles. Let $H \subseteq SE(3)$ denote the configuration space of the hand when the fingers are fully open. We will refer to a configuration $h \in H$ as simply a "hand". Let $B(h) \subseteq W$ denote the volume occupied by the hand in configuration $h \in H$, when the fingers are fully open.

**Definition 2.** *An* **antipodal hand** *is a pose of the hand, $h \in H$, such that the hand is not in collision with any objects or obstacles, $B(h) \cap \mathcal{O} = \emptyset$, and at least one pair of antipodal contacts will be formed when the fingers close such that the line connecting the two contacts is parallel to the direction of finger motion.*

Algorithm 1 illustrates at a high level our algorithm for detecting antipodal hands. It takes a point cloud, $\mathcal{C} \subseteq \mathbb{R}^3$, and a geometric model of the robot hand as input and produces as output a set of hands, $\mathcal{H} \subseteq H$, that are predicted to be antipodal. There are two main steps. First, we sample a set of hand hypotheses. Then, we classify each hypothesis as an antipodal hand or not. These steps are described in detail in the following sections.

**Algorithm 1** Detect_Antipodal_Hands

**Input:** a point cloud, $\mathcal{C}$, and hand parameters, $\theta$
**Output:** antipodal hands, $\mathcal{H}$

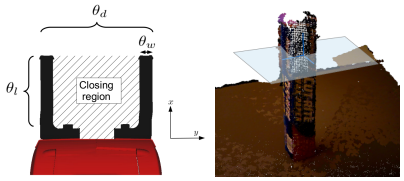
1: $\mathcal{H}_{hyp} = Sample\_Hands(\mathcal{C})$
2: $\mathcal{H} = Classify\_Hands(\mathcal{H}_{hyp})$

## 3 Sampling Hands

A key part of our algorithm is the approach to sampling from the space of hand hypotheses. A naive approach would be to sample directly from $H \subseteq SE(3)$. Unfortunately, this would be immensely inefficient because $SE(3)$ is a 6-DOF space and many hands sampled this way would be far away from any visible parts of the point cloud. Instead, we define a lower-dimensional sample space constrained by the geometry of the point cloud.

### 3.1 Geometry of the Hand and the Object Surface

Before describing the sample space, we quantify certain parameters related to the grasp geometry. We assume the hand, $h \in H$, is a parallel jaw gripper comprised of two parallel fingers each modeled as a rectangular prism that moves parallel to a common plane. Let $\hat{a}(h)$ denote a unit vector orthogonal to this plane. The hand is fully specified by the parameter vector $\theta = (\theta_l, \theta_w, \theta_d, \theta_t)$ where $\theta_l$ and $\theta_w$ denote the length and width of the fingers; $\theta_d$ denotes the distance between the fingers when fully open; and $\theta_t$ denotes the thickness of the fingers (orthogonal to the page in Figure 2 (a)). Define the **closing region**, $R(h) \subseteq \mathcal{W}$, to be the volumetric region swept out by the fingers when they close. Let $r(h) \in R(h)$ denote an arbitrary reference point in the closing region. Define the **closing plane**, $C(h)$, to be the subset of the plane that intersects $r(h)$, is orthogonal to $\hat{a}(h)$, and is contained within $R(h)$:




(a) (b)

**Fig. 2.** (a) hand geometry. (b) cutting plane geometry.

$$C(h) = \{p \in R(h) | (p - r(h))^T \hat{a}(h) = 0\}.$$

We also introduce some notation related to the differential geometry of the surfaces we are grasping. Recall that each point on a differentiable surface is associated with a surface normal and two principal curvatures where each principal curvature is associated with a principal direction. The surface normal and the two principal directions define an orthogonal basis known as a Darboux frame [2]. The Darboux frame at point $p \in \mathcal{C}$ will be denoted: $F(p) = (\hat{n}(p)\ (\hat{a}(p) \times \hat{n}(p))\ \hat{a}(p))$, where $\hat{n}(p)$ denotes the unit surface normal and $\hat{a}(p)$ denotes the direction of minimum principal curvature at point $p$. Define the **cutting plane** to be the plane orthogonal to $\hat{a}(p)$ that passes through $p$ (see Figure 2 (b)). Since we are dealing with point clouds, it is not possible to measure the Darboux frame exactly at each point. Instead, we estimate the surface normal and principle directions over a small neighborhood. We fit a quadratic function over the points contained within a small ball (3 cm radius in our experiments) using Taubin's method [18,19] and use that to calculate the Darboux frame [3].

[2] Any frame aligned with the surface normal is a Darboux frame. Here we restrict consideration to the special case where it is also aligned with the principal directions.

[3] Taubin's method is an analytic solution that performs this fit efficiently by solving a generalized Eigenvalue problem on two $10 \times 10$ matrices [18]. In comparison to using first order estimates of surface normal and curvature, the estimates derived from this quadratic are more robust to local surface discontinuities.

### 3.2 Hand Sample Set

We want a set that contains many antipodal hands and from which it is easy to draw samples. The following conditions define the set $\mathcal{H}$. First, for every hand, $h \in \mathcal{H}$:

**Constraint 1.** *The body of the hand is not in collision with the point cloud:* $B(h) \cap \mathcal{C} = \emptyset$,

Furthermore, there must exist a point in the cloud, $p \in \mathcal{C}$, such that:

**Constraint 2.** *The hand closing plane contains p:* $p \in C(h)$.

**Constraint 3.** *The closing plane of the hand is parallel to the cutting plane at p:* $\hat{a}(p) = \hat{a}(h)$.

These three constraints define the following set of hands:

$$\mathcal{H} = \cup_{p \in \mathcal{C}} H(p),\ H(p) = \{h \in H | p \in C(h) \wedge \hat{a}(p) = \hat{a}(h) \wedge B(h) \cap \mathcal{C} = \emptyset\}. \quad (1)$$

Constraint 3 is essentially a heuristic that limits the hand hypotheses that our algorithm considers. While this eliminates from consideration many otherwise good grasps, it is a practical way to focus detection on likely candidates. One motivation behind this constraint is that humans prefer grasps for which the wrist is oriented orthogonally to one of the object's principal axes [1]. Moreover, it is easy to sample from $\mathcal{H}$ by: 1) sampling a point, $p \in \mathcal{C}$, from the cloud; 2) sampling one or more hands from $H(p)$. Notice that for each $p \in \mathcal{C}$, $H(p)$ is three-DOF because we have constrained two DOF of orientation and one DOF of position. This means that $\mathcal{H}$ is much smaller than $H$ and it can therefore be covered by many fewer samples.

**Algorithm 2** Sample_Hands

**Input:** point cloud, $\mathcal{C}$, hand parameters, $\theta$
**Output:** grasp hypotheses, $\mathcal{H}$

1: $\mathcal{H} = \emptyset$
2: Preprocess $\mathcal{C}$ (voxelize; workspace limits; *etc.*)
3: **for** i = 1 to n **do**
4: Sample $p \in \mathcal{C}$ uniformly randomly
5: Calculate $\theta_d$-ball about $p$: $N(p) = \{q \in \mathcal{C} : \|p - q\| \leq \theta_d\}$
6: Estimate local Darboux frame at $p$: $F(p) = Estimate\_Darboux(N(p))$
7: $H = Grid\_Search(F(p), N(p))$
8: $\mathcal{H} = \mathcal{H} \cup H$
9: **end for**

The sampling process is detailed in Algorithm 2. First, we preprocess the point cloud, $\mathcal{C}$, in the usual way by voxelizing (we use voxels 3mm on a side in our experiments) and applying workspace limits (Step 2). Second, we iteratively sample a set of $n$ points ($n$ is between 4000 and 8000 in our experiments) from the cloud (Step 4). For each point, $p \in \mathcal{C}$, we calculate a neighborhood, $N(p)$,

in the $\theta_d$-ball around $p$ (using a KD-tree, Step 5). The next step is to estimate the Darboux frame at $p$ by fitting a quadratic surface using Taubin's method and calculating the surface normal and principal curvature directions (Step 6). Next, we sample a set of hand configurations over a coarse two-DOF grid in a neighborhood about $p$. Let $h_{x,y,\phi}(p) \in H(p)$ denote the hand at position $(x, y, 0)$ with orientation $\phi$ with respect to the Darboux frame, $F(p)$. Let $\Phi$ denote a discrete set of orientations (8 in our implementation). Let $X$ denote a discrete set of hand positions (20 in our implementation). For each hand configuration $(\phi, x) \in \Phi \times X$, we calculate the hand configuration furthest along the $y$ axis that remains collision free: $y^* = \max_{y \in Y}$ such that $B(h_{x,y,\phi}) \cap N = \emptyset$, where $Y = [-\theta_d, \theta_d]$ (Step 3). Then, we check whether the closing plane for this hand configuration contains points in the cloud (Step 4). If it does, then we add the hand to the hypotheses set (Step 5).

**Algorithm 3** Grid_Search

**Input:** neighborhood point cloud, $N$; Darboux frame, $F$
**Output:** neighborhood grasp hypotheses, $H$

1: $H = \emptyset$
2: **for all** $(\phi, x) \in \Phi \times X$ **do**
3: Push hand until collision: $y^* = \max_{y \in Y}$ such that $B(h_{\phi,x,y}) \cap N = \emptyset$
4: **if** closing plane not empty: $C(h_{\phi,x,y^*}) \cap N \neq \emptyset$ **then**
5: $H = H \cup h_{\phi,x,y^*}$
6: **end if**
7: **end for**

### 3.3 Grasping Results

Interestingly, our experiments indicate that this sampling method by itself can be used to do grasping. In Algorithm 1, the sampling process is followed by the grasp classification process described in the next section. However, if we omit classification, implicitly assuming that all grasp hypotheses are true grasps, we obtain a surprisingly high grasp success rate of approximately 73% (the column labeled *NC, 2V* in Figure 7). The experimental context of this result is described in Section 5. Essentially, we cluster the sampled hands and use a heuristic grasp selection strategy to choose a grasp to execute (see Section 5.1). This result is surprising because the sampling constraints (Constraints 1–3) encode relatively simple geometric conditions. It suggests that these sampling constraints are an important part of our overall grasp success rates.

## 4 Classifying Hand Hypotheses

After generating hand hypotheses, the next step is to classify each of those hypotheses as antipodal or not. The simplest approach would be to infer object surface geometry from the point cloud and then check which hands satisfy Definition 2. Unfortunately, since most real-world point clouds are partial, many

hand hypotheses will fail this check simply because all relevant object surfaces were not visible to a sensor. Instead, we infer which hypotheses are likely to be antipodal using machine learning (*i.e.* classification).

### 4.1 Labeling Grasp Hypotheses

Many approaches to grasp point detection require large amounts of training data where humans have annotated images with good grasp points [16,8,12,4,5,7]. Unfortunately, obtaining these labels is challenging because it can be hard for human labelers to predict what object surfaces in a scene might be graspable for a robot. Instead, our method automatically labels a set of training images by checking a relaxed version of the conditions of Definition 2.

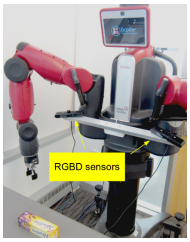
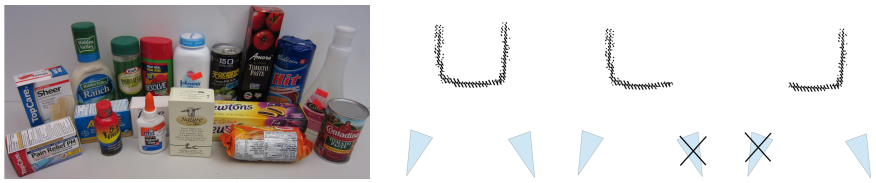



**Fig. 3.** Our robot has stereo RGBD sensors.

In order to check whether a hand hypotheses, $h \in H$, is antipodal, we need to determine whether an antipodal pair of contacts will be formed when the hand closes. Let $\hat{f}(h)$ denote the direction of closing of one finger. (In a parallel jaw gripper, the other finger closes in the opposite direction). When the fingers close, they will make first contact with an extremal pair of points, $s_1, s_2 \in R(h)$ such that $\forall s \in R(h), s_1^T \hat{f}(h) \geq s^T \hat{f}(h) \wedge s_2^T \hat{f}(h) \leq s^T \hat{f}(h)$. An antipodal hand requires two such extremal points to be antipodal and for the line connecting the points to be parallel to the direction of finger closing. In practice, we relax this condition slightly as follows. First, rather than checking for extremal points, we check for points that have a surface normal parallel to the direction of closing. This is essentially a first-order condition for an extremal point that is more robust to outliers in the cloud. The second way that we relax Definition 2 is to drop the requirement that the line connecting the two contacts be parallel to the direction of finger closing and to substitute a requirement that at least $k$ points are found with an appropriate surface normal. Again, the intention here is to make detection more robust: if there are at least $k$ points near each finger with surface normals parallel to the direction of closing, then it is likely that the line connecting at least one pair will be nearly parallel to the direction of finger closing. In summary, we check whether the following definition is satisfied:

**Definition 3.** *A hand, $h \in H$, is* **near antipodal** *for thresholds $k \in \mathbb{N}$ and $\theta \in [0, pi/2]$ when there exist $k$ points $p_1, \ldots, p_k \in R(h) \cap \mathcal{C}$ such that $\hat{n}(p_i)^T \hat{f}(h) \geq \cos\theta$ and $k$ points $q_1, \ldots, q_k \in R(h) \cap \mathcal{C}$ such that $\hat{n}(q_i)^T \hat{f}(h) \leq -\cos\theta$.*

of grasps When Definition 3 is satisfied, then we label the corresponding hand a positive instance. Note that in order to check for this condition, it is necessary to register at least two point clouds produced by range sensors that have observed the scene from different perspectives (Figure 3). This is because we need to "see" two nearly opposite surfaces on an object. Even then, many antipodal hands will not be identified as such because only one side of the object is visible.



(a) (b) (c) (d)

**Fig. 5.** (a) training set comprised of 18 objects. (b-d) illustration of the three grasp hypotheses images incorporated into the training set per hand. The blue triangles at the bottom denote positions of the two range sensors. (c-d) illustrate training images created using data from only one sensor.

These "indeterminate" hands are omitted from the training set. In some cases, it is possible to verify that a particular hand is *not* antipodal by checking that there are fewer than $k$ points in the hand closing region that satisfy either of the conditions of Definition 3. These hands are included in the training set as negative examples. This assumes that the closing region of every sampled hand hypothesis is at least partially visible to a sensor. If there are fewer than $k$ satisfying points, then Definition 3 would not be satisfied even if the opposite side of an object was observed. In our experiments, we set the thresholds $k = 6$ and $\theta = 20$ degrees. However, our qualitative results are not terribly sensitive to these exact numbers. In general, it might be necessary to tune these parameters (especially $k$) with respect to the number of points that can be expected to be found on a graspable object and the accuracy of the surface normal estimates.

### 4.2 Feature Representation

In order to classify hand hypotheses, a feature descriptor is needed. Specifically, for a given hand $h \in H$, we need to encode the geometry of the points contained within the hand closing region, $\mathcal{C} \cap R(h)$. A variety of relevant descriptors have been explored in the literature [15,20]. In our case, we achieve good performance using a simple descriptor based on HOG features. For a point cloud, $\mathcal{C}$, a two dimensional image of the closing region is created by projecting the points $\mathcal{C} \cap R(h)$ onto the hand closing plane: $I(\mathcal{C}, h) = S_{12} F(h)^T (N \cap C(h))$, where $S_{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ selects the first two rows of $F(h)^T$. We call this the **grasp hypothesis image**. We encode it using the HOG descriptor, $HOG(I(\mathcal{C}, h))$. In our implementation, we chose a HOG cell size such that the grasp hypothesis image was covered by $10 \times 12$ cells with a standard $2 \times 2$ block size.

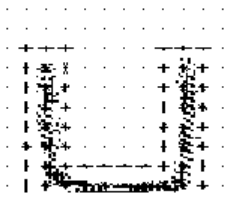


**Fig. 4.** HOG feature representation of a hand hypothesis for the box shown in Figure 2 (b).

### 4.3 Creating the Training Set

In order to create the training set, we obtain a set of objects that have local geometries similar to what might be expected in the field. In our work, we selected the set of 18 objects shown in Figure 5 (a). Each object was placed in front of the robot in two configurations: one upright configuration and one on its side. For each configuration (36 configurations total), let $\mathcal{C}_1$ and $\mathcal{C}_2$ denote the voxelized point clouds obtained from each of the two sensors, respectively, and let $\mathcal{C}_{12} = \mathcal{C}_1 \cup \mathcal{C}_2$ denote the registered two-view cloud.

The training data is generated as follows. First, we extract hand hypotheses from the registered cloud, $\mathcal{C}_{12}$ using the methods of Section 3. Second, for each $h \in H$, we determine whether it is a positive, negative, or indeterminate by checking the conditions of Definition 3. Indeterminate hands are discarded from training. Third, for each positive or negative hand, we extract three feature descriptors: $HOG(I(\mathcal{C}_1, h))$, $HOG(I(\mathcal{C}_2, h))$, and $HOG(I(\mathcal{C}_{12}, h))$. Each descriptor is given the same label and incorporated into the training set. Over our 18 object training set, this procedure generated approximately 6500 positive and negative labeled examples that were used to train an SVM. We only did one round of training using this single training set.

The fact that we extract three feature descriptors per hand in step three above is important because it helps us to capture the appearance of partial views in the training set. Figure 5 (b-d) illustrates the three descriptors for an antipodal hand. Even though the closing region of this hand is relatively well observed in $\mathcal{C}_{12}$, the fact that we incorporate $HOG(I(\mathcal{C}_1, h))$ and $HOG(I(\mathcal{C}_2, h))$ into the dataset means that we are emulating what *would* have been observed if we only had a partial view. This makes our method much more robust to partial point cloud information.

### 4.4 Cross Validation

We performed cross validation on a dataset derived from the 18 training objects shown in Figure 5 (a). For each object, we obtained a registered point cloud for two configurations (total of 36 configurations). Following the procedure described in this section, we obtained 6500 labeled features with 3405 positives and 3095 negatives. We did 10-fold cross validation on this dataset using an SVM for the various Gaussian and polynomial kernels available in Matlab. We obtained 97.8% accuracy using a degree-three polynomial kernel and used this kernel in the remainder of our experiments. In the cross validation experiment described above, the folds were random across the labeled pairs in the dataset. This does not capture the effects of experiencing novel objects or the expected performance

**Fig. 6.** The 30 objects in our test set.

when only single-view point clouds are available. Therefore, we did the following. First, we trained the system using the degree-three polynomial kernel on the 6500 labeled examples as described above. Then, we obtained additional single-view point clouds for each of the 30 novel test objects shown in Figure 6 (each object was presented in isolation) for a total of 122 single-view points clouds. We used the methods described in this section to obtain ground-truth for this dataset. This gave us a total of 7250 labeled single-view hypotheses on novel objects with 1130 positives and 6120 negatives. We obtained 94.3% accuracy on this dataset. The fact that we do relatively well in these cross validation experiments using a relatively simple feature descriptor and without mining hard negatives suggests that our approach to sampling hands and creating the grasp hypothesis image makes the grasp classification task easier than it is in approaches that do not use this kind of structure [16,8,4,5,12].

## 5 Robot Experiments

We evaluated the performance of our algorithms using the Baxter robot from Rethink Robotics. We explore two experimental settings: when objects are presented to the robot in isolation and when objects are presented in a dense clutter scenario. We use the Baxter right arm equipped with the stock two-finger Baxter gripper. A key constraint of the Baxter gripper is the limited finger stroke: each finger has only 2 cm stroke. In these experiments, we adjust the finger positions such that they are 3 cm apart when closed and 7 cm apart when open. This means we cannot grasp anything smaller than 3 cm or larger than 7 cm. We chose each object in the training and test sets so that it could be grasped under these constraints. Two-view registered point clouds were created using Asus Xtion Pro range sensors (see Figure 3). We implemented our algorithm in C++ on an Intel i7 3.5GHz system (four physical CPU cores) with 16GB of system memory. On average, learning the SVM model on the 18 object training set shown in Figure 5(a) takes about five minutes, while online grasp detection and selection given a single point cloud takes about 2.7s (with 4000 hand samples). It should be possible for anyone with a Baxter robot and the appropriate depth sensors to replicate any of these experiments by running our ROS package at `http://wiki.ros.org/agile_grasp`.

### 5.1 Grasp Selection

Since our algorithm typically finds tens or hundreds of potential antipodal hands, depending upon the number of objects in the scene, it is necessary to select one to execute. One method might be to select a grasp on an object of interest. However, in this paper, we ignore object identity and perform any feasible grasp. We choose a grasp to attempt as follows. First, we sparsify the set of grasp choices by clustering antipodal hands based on distance and orientation. Grasp hypothesis that are nearby each other and that are roughly aligned in orientation are grouped together. Each cluster must be composed of a specified minimum

| Object | number of poses | Succ. Rate A, 2V | | number of poses | Success Rate NC, 1V | NC, 2V | SVM, 1V | SVM, 2V |
|---|---|---|---|---|---|---|---|---|
| Plush drill | 3 | 100.00% | | 6 | 50.00% | 66.67% | 100.00 | 66.67% |
| Black pepper | 3 | 100.00% | | 8 | 62.5% | 62.50% | 75.00 | 100.00% |
| Dremel engraver | 3 | 100.00% | | 6 | 33.33% | 50.00% | 66.67 | 100.00% |
| Sand castle | 3 | 100.00% | | 6 | 50.00% | 33.33% | 83.33 | 83.33% |
| Purple ball | 0 | NA | | 6 | 66.67% | 100.00% | 83.33 | 100.00% |
| White yarn roll | 3 | 100.00% | | 8 | 87.50% | 87.50% | 87.50 | 75.00% |
| Odor protection | 0 | NA | | 8 | 50.00% | 87.50% | 87.50 | 75.00% |
| Neutrogena box | 3 | 66.67% | | 8 | 25.00% | 87.50% | 87.50 | 87.50% |
| Plush screwdriver | 3 | 100.00% | | 6 | 83.33% | 87.50% | 83.33 | 100.00% |
| Toy banana box | 3 | 100.00% | | 8 | 100% | 83.33% | 87.50 | 75.00% |
| Rocket | 3 | 100.00% | | 8 | 50.00% | 87.50% | 100.00 | 87.50% |
| Toy screw | 3 | 100.00% | | 6 | 100.00% | 100.00% | 83.33 | 100.00% |
| Lamp | 3 | 100.00% | | 8 | 62.50% | 83.33% | 87.50 | 87.50% |
| Toothpaste box | 3 | 66.67% | | 8 | 87.50% | 100.00% | 87.50 | 87.50% |
| White squirt bottle | 3 | 66.67% | | 8 | 25.00% | 12.50% | 75.00 | 87.50% |
| White rope | 3 | 100.00% | | 6 | 66.67% | 83.33% | 83.33 | 100.00% |
| Whiteboard cleaner | 3 | 100.00% | | 8 | 62.50% | 75.00% | 100.00 | 100.00% |
| Toy train | 0 | NA | | 8 | 87.50% | 100.00% | 87.50 | 100.00% |
| Vacuum part | 3 | 100.00% | | 6 | 33.33% | 66.67% | 100.00 | 83.33% |
| Computer mouse | 0 | NA | | 6 | 33.33% | 33.33% | 66.67 | 83.33% |
| Vacuum brush | 1 | 100% | | 6 | 50.00% | 83.33% | 66.67 | 50.00% |
| Lint roller | 3 | 100.00% | | 8 | 75.00% | 75.00% | 87.50 | 100.00% |
| Ranch seasoning | 3 | 100.00% | | 8 | 50.00% | 75.00% | 100.00 | 100.00% |
| Red pepper | 3 | 100.00% | | 8 | 75.00% | 75.00% | 100.00 | 100.00% |
| Crystal light | 3 | 100.00% | | 8 | 25.00% | 37.50% | 75.00 | 75.00% |
| Red thread | 3 | 100.00% | | 8 | 75.00% | 100.00% | 100.00 | 100.00% |
| Kleenex | 3 | 100.00% | | 6 | 33.33% | 33.33% | 83.33 | 83.33% |
| Lobster | 3 | 66.67% | | 6 | 16.67% | 83.33% | 66.67 | 83.33% |
| Boat | 3 | 100.00% | | 6 | 83.33% | 100.00% | 83.33 | 100.00% |
| Blue squirt bottle | 2 | 100% | | 8 | 25.00% | 50.00% | 75.00 | 62.50% |
| **Average** | | **94.67%** | | | **57.50%** | **72.92%** | **85.00%** | **87.78%** |

**Fig. 7.** Single object experimental results. Algorithm variations are denoted as: A for antipodal grasps (see Section 4.1), NC for sampling without grasp classification (see Section 3), and SVM for our full detection system.

number of constituent grasps. If a cluster is found, then we create a new grasp hypothesis positioned at the mean of the cluster and oriented with the "average" orientation of the constituent grasps. 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 criteria: 1) distance from joint limits (a piecewise function that is zero far from the arm joint limits and quadratic nearby the limits); 2) distance from hand joint limits (zero far from the limits and quadratic nearby limits); 3) workspace distance traveled by the hand starting from a fixed pre-grasp arm configuration. 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 the one that minimizes Criterion #3 as the grasp to be executed by the robot.

### 5.2 Objects Presented in Isolation

We performed a series of experiments to evaluate how well various parts of our algorithm perform in the context of grasping each of the 30 test set objects (Figure 6). Each object was presented to the robot in isolation on a table in front of the robot. We characterize three variations on our algorithm:

1. **No Classification:** We assume that all hand hypotheses generated by the sampling algorithm (Algorithm 2) are antipodal and pass all hand samples directly to the grasp selection mechanism without classification as described in Section 5.1.
2. **Antipodal:** We classify hand hypotheses by evaluating the conditions of Definition 3 directly for each hand and pass the results to grasp selection.
3. **SVM:** We classify hand hypotheses using the SVM and pass the results to grasp selection. The system was trained using the 18-object training set as described in Section 4.4.

In all scenarios, a grasp trial was considered a success only when the robot successfully localized, grasped, lifted, and transported the object to a box on the side of the table. We evaluate *No Classification* and *SVM* for single-view and two-view registered points clouds over 214 grasps of the 30 test objects. Each object was placed in between 6 and 8 systematically different orientations relative to the robot.

Figure 7 shows the results. The results for *No Classification* are shown in columns *NC, 1V* and *NC, 2V*. Column *NC, 1V* shows that with a point cloud created using only one depth sensor, using the results of sampling with no additional classification results in an average grasp success rate of 58%. However, as shown in Column *NC, 2V*, it is possible to raise this success rate to 73% just by adding a second depth sensor and using the resulting two-view registered cloud. The fact that we obtain a grasp success rate as high as 73% here is surprising considering that the sample strategy employs rather simple geometric constraints. This suggests that even simple geometric constraints can improve grasp detection significantly. The results for *Antipodal* are shown in the column labeled *A, 2V*. We did not evaluate this variation for a one-view cloud because a two-view cloud is needed for Definition 3 to find any near antipodal hands. Compared to the other two approaches, *Antipodal* finds relatively few positives. This is because this method needs to "see" two sides of a potential grasp surface in order to verify the presence of a grasp. As a result, we were only able to evaluate this method over three poses per object instead of six or eight. In fact, *Antipodal* failed to find any grasps at all for four of the 30 objects. Although *Antipodal* appears to be effective (94.7% grasp success rate), it is not very useful in practice since it works only for a small subset of possible object orientations. The results for *SVM* are shown in columns *SVM, 1V* and *SVM, 2V* (results for one-view and two-view point clouds, respectively). Interestingly, there is not much advantage here to adding a second depth camera: we achieve an 85.0% success rate with a one-view point cloud and an 87.8% success rate with a two-view registered cloud. Drilling down into these numbers, we find the following three major causes of grasp failure: 1) approximately 5.6% of the grasp failure rate in both scenarios is due to collisions between the gripper and the object caused by arm calibration errors (*i.e.* an inaccurate kinematic model that introduces error into the calculation of forward/inverse kinematics) or collisions with observed or unobserved parts of the environment; 2) approximately 3.5 % of the objects were dropped after a successful initial grasp; 3) approximately 2.3% of grasp failures in the

two-view case (3.7% in the one view case) were caused by perceptual failures by our algorithm. The striking thing about the causes of failure listed above is that they are not all perceptual errors: if we want to improve beyond the 87.8% success rate, we need to improve performance in multiple areas.

In the experiments described above, we eliminated seven objects from the test set because they were hard to see with our depth sensor (Asus Primesense) due to specularity, transparency, or color. We characterized grasp performance for these objects separately by grasping each of these objects in eight different poses (total of 56 grasps over all seven objects). Using *SVM*, we obtain a 66.7% grasp success rate using a single-view point cloud and a 83.3% grasp success rate when a two-view cloud is used. This result suggests: 1) our 87.8% success rate drops to 83% for hard-to-see objects; 2) creating a more complete point cloud by adding additional sensors is particularly important in non-ideal viewing conditions.

**Fig. 8.** Hard-to-see objects.

### 5.3 Objects Presented in Dense Clutter

We also characterized our algorithm in dense clutter as illustrated in Figure 9. We created a test scenario where ten objects are piled together in a shallow box. We used exactly the same algorithm (i.e. *SVM*) in this experiment as in the isolated object experiments. We used a two-view registered point cloud in all cluttered scenarios. The 27 objects used in this experiment are a subset of the 30 objects used in the single object experiments. We eliminated the computer mouse and the engraver because they have cables attached to them that can get stuck in the clutter. We also removed the vacuum brush because the brush part cannot be grasped by the Baxter gripper in some configurations due to the 3–7 cm aperture limits. At the beginning of each run, we randomly selected 10 out of the 27 objects and placed them in a small rectangular container. We then shook the container to mix up the items and emptied it into the shallow box on top of the table. We excluded all runs where the sandcastle landed upside down because the Baxter gripper cannot grasp it in that configuration. A run was terminated when three consecutive localization failures occurred. In total, we performed 10 runs of this experiment.

Over all 10 runs of this experiment, the robot performed 113 grasps. On average, it succeeded in removing 85% of the objects from each box. The remaining objects were not grasped because the system failed to localize a grasp point three times in a row. Over all grasp attempts, 73% succeeded. The 27% failure rate breaks down into the following major failure modes: 3% due to kinematic modelling errors; 9% due to perceptual failures caused by our algorithm; 4% due to dropped objects following a successful grasp; and 4% due to collision with the environment. In comparison with the isolation results, these results have a significantly higher perceptual failure rate. We believe this is mainly due to the extensive occlusions in the clutter scenario.

(a) (b)

**Fig. 9.** Dense clutter scenario. (a) RGB image. (b) Output of our algorithm.

## 6 Discussion

This paper proposes a new approach to localizing grasp poses of novel objects presented in clutter. The main contributions are: 1) a method of using grasp geometry to generate a set of hypotheses that focus grasp detection on relevant areas of the search space; 2) a method of using grasp geometry to label a training set automatically, thereby enabling the creation of a large training set grounded in grasp mechanics. As a result of these contributions, our method stands out from the existing work (for example [4,12,3,7,11]) in a couple of different ways. First, our method can detect 6-DOF hand *poses* from which a grasp is expected to succeed rather than detecting grasp *points* (typically 3-DOF) in a depth image or heightmap such as in [4,12]. Second, our method of automatic training set generation should enable us to train better classifiers because we can generate as much training data as we want and reduce label noise because labels are generated based on objective mechanical conditions for a grasp. It should also be noted that our grasp hypothesis generation mechanism might be used independently of the grasp classification strategy. It is striking that the proposal mechanism alone (without any classification but with outlier removal) can yield a 73% grasp success rate when grasping objects presented in isolation. Essentially, this sample set constitutes a proposal distribution that should boost the performance of any classifier. Finally, the fact that we document a drop in grasp success rate from 87.8% for objects presented in isolation to 73% for objects presented in clutter suggests that clutter is a significantly more difficult problem that needs to be studied more closely.

### Acknowledgements

This work was supported in part by NASA under Grant No. NNX13AQ85G, ONR under Grant No. N000141410047, and the NSF under Grant No. 1427081.

## References

1. R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *IEEE Intl Conf. on Robots and Automation*, pages 2294–2301. IEEE, 2010.

2. Lillian Chang, Joshua R Smith, and Dieter Fox. Interactive singulation of objects from a pile. In *IEEE Int'l Conference on Robotics and Automation*, pages 3875–3882. IEEE, 2012.
3. Renaud Detry, Carl Henrik Ek, Marianna Madry, and Danica Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE Int'l Conference on Robotics and Automation*, 2013.
4. 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.
5. David Fischinger, Markus Vincze, and Yun Jiang. Learning grasps for unknown objects in cluttered scenes. In *IEEE Int'l Conference on Robotics and Automation*, pages 609–616. IEEE, 2013.
6. J. Glover and S. Popovic. Bingham procrustean alignment for object detection in clutter. In *IEEE Int'l Conf. on Intelligent Robot Systems*, 2013.
7. 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.
8. 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.
9. 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.
10. 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.
11. Oliver Kroemer, Emre Ugur, Erhan Oztop, and Jan Peters. A kernel-based approach to direct action perception. In *IEEE Int'l Conf. on Robots and Automation*, pages 2605–2610. IEEE, 2012.
12. I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. In *Robotics: Science and Systems*, 2013.
13. Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
14. V. Nguyen. Constructing force-closure grasps. In *IEEE Int'l Conf. Robotics Automation*, volume 3, pages 1368–1373, April 1986.
15. R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Int'l Conf. on Robots and Automation*, 2009.
16. A. Saxena, J. Driemeyer, and A. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(4):157, 2008.
17. A. Sudsang and J. Ponce. New techniques for computing four-finger force-closure grasps of polyhedral objects. In *IEEE Int'l Conf. Robotics Automation*, volume 2, pages 1355–1360, May 1995.
18. 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.
19. A. ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In *Int'l Symposium on Experimental Robotics*, 2014.
20. 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.