Viewpoint Selection for Grasp Detection

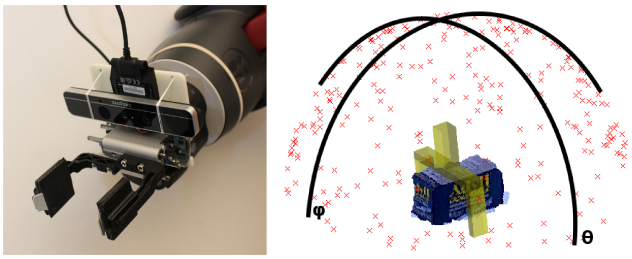
Marcus Gualtieri and Robert Platt*

***Abstract*— In grasp detection, the robot estimates the position and orientation of potential grasp configurations directly from sensor data. This paper explores the relationship between viewpoint and grasp detection performance. Specifically, we consider the scenario where the approximate position and orientation of a desired grasp is known in advance and we want to select a viewpoint that will enable a grasp detection algorithm to localize it more precisely and with higher confidence. Our main findings are that the right viewpoint can dramatically increase the number of detected grasps and the classification accuracy of the top-n detections. We use this insight to create a viewpoint selection algorithm and compare it against a random viewpoint selection strategy and a strategy that views the desired grasp head-on. We find that the head-on strategy and our proposed viewpoint selection strategy can improve grasp success rates on a real robot by 8% and 4%, respectively. Moreover, we find that the combination of the two methods can improve grasp success rates by as much as 12%.**

I. Introduction

Grasp detection has become an important framework for perception for grasping [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. We define *grasp detection* as any system that localizes grasps directly in visual sensory data without reference to prior geometric models of objects. Figure 2 (c) shows an example. Using only the visible point cloud and without a model of the object to be grasped, the system locates hand configurations from which the object can be grasped (shown as yellow grippers in the figure). Each gripper denotes the 6-DOF pose of a robotic hand from which a grasp would be feasible if the gripper fingers were to close. Relative to more traditional approaches to perception for grasping based on estimating the exact pose of the object to be grasped, grasp detection can be very effective for novel objects in cluttered environments. Although grasp detection works well in general, our prior work suggests the quality and amount of sensor data available could have a significant impact on performance [9]. In particular, when given relatively complete point cloud data generated using InfiniTAM, a metric SLAM software [12], grasp detection was able to produce a 93% grasp success rate for novel object grasping in dense clutter. However, grasp success rates dropped to 84% for the same algorithm and the same robot when using data produced using two, fixed depth sensors. This inspired us to look at the relationship between viewpoint and the performance of a given grasp detection algorithm.

*College of Computer and Information Science, Northeastern University, Boston, Massachusetts, USA, mgualti@ccs.neu.edu. This work has been supported in part by the National Science Foundation through IIS-1427081, NASA through NNX16AC48A and NNX13AQ85G, and ONR through N000141410047.



(a) (b)

Fig. 1. (a) The eye-in-hand configuration used for our robot experiments. (b) The viewpoint selection problem. From which viewpoint should this object be viewed in order to detect a grasp nearby the one shown?

In this paper, we ask whether it is possible to affect the accuracy of grasp detection by selecting the right viewpoint. Specifically, we consider the scenario where the approximate position and orientation (pose) of the desired grasp is known in advance and we want to select a viewpoint in order to localize it more accurately and with higher confidence. For example, imagine the robot has already taken an RGBD image of the environment from an arbitrary viewpoint and identified an object of interest. The robot *thinks* it knows how it wants to grasp the object, but it would like to take a second look to improve its confidence. Figure 1 (b) illustrates this scenario. The robot thinks the grasp is located as shown in the image, but it would like to select a viewpoint from which to take a second look and "confirm" the detection.

Our main finding is that viewpoint most significantly affects the number of grasps detected. With the right viewpoint, the system can increase the number of detected grasps by a factor of 6 (Table I). This is important because a larger number of grasp detections can give the robot more choice about exactly which grasp it chooses to execute. Also, it enables the robot to be more selective about which grasps it accepts, thereby effectively increasing the accuracy for the most highly ranked n grasp detections. Based on these findings, we propose a "smart" viewpoint selection algorithm and compare it offline against random viewpoint selection and a baseline that always selects a head-on view. These experiments show that the smart viewpoint selection strategy outperforms the random strategy and that it can also outperform the head-on strategy for some classes of objects. We also evaluate the method in terms of grasp success rates in a real robotic system. Here, we used the head-on view only to "align" the desired grasp with the point cloud. We find that the smart viewpoint outperforms random by an average of 4% while the head-on alignment view outperforms random by 8%. By combining both methods (choosing the smart

viewpoint, detecting grasps, and then choosing a head-on view for the grasp detected from the smart viewpoint), we outperform the random baseline by an average of 12%.

II. Background and Related Work

A. Active vision

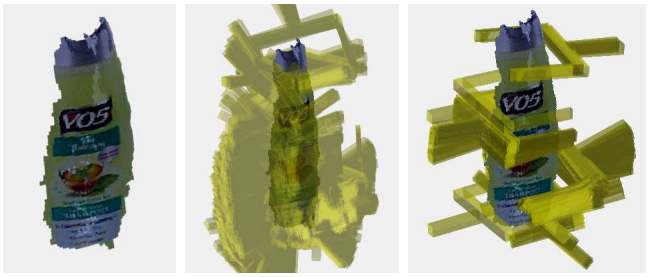
This work falls broadly into the category of research called "active vision", which is defined to be any scenario where the robot applies a strategy for sensor placement/configuration to perform its task [13]. Chen *et al.* provide a broad (although now somewhat dated) survey of the subject [13]. Our paper is more specifically related to the idea of planning sensor placements for the task of object recognition. Roy *et al.* survey works that benefit object recognition by purposive sensor placements [14], and Velez *et al.* plan viewing trajectories to improve the performance of an off-the-shelf object detector [15]. In contrast, our task is not to recognize object instances but instead to recognize grasps that are likely to succeed when executed; although, we expect some of the ideas could extend to a more general class of detection problems.

An issue with any viewpoint selection method is deciding a metric to use for viewpoint quality. Chen *et al.* suggest that the best metric to use is likely to be task-dependent [13]. On the other hand, general trends can be observed in the literature. One idea is to try to increase the amount of information of the scene by maximizing over Shannon entropy [16], [17], KL divergence [18], or Fisher information [19]. These methods all require specifying a probability distribution over which to compute the information metric. In contrast, we directly evaluate the performance of the grasp detection system from various viewpoints and store the result in a database, which could be viewed as a non-parametric, nearest-neighbor approach [20]. We argue that this approach is simpler to implement and directly applicable to the task at hand.

In this work we restrict our attention to single views; although, constructing a scene from multiple views can sometimes be a powerful approach. We and others have explored this with significant benefit to grasp detection performance [21], [9], but the primary difficulty faced with this is that the ICP-based SLAM algorithms do not register multiple views well in near-field, uncluttered scenes.

B. Grasp detection

In order to understand the problem of viewpoint selection for grasp detection, it is important to understand grasp detection itself. In this paper, we use the grasp detector described in our prior work [9] (and, for comparison, a modification of that detector designed to be similar to the detector proposed by [7]). This system takes point clouds as input and produces predicted grasp poses in $SE(3)$ [1] as output. Each grasp pose output by the system is predicted to be a force closure grasp (i.e. a *positive* grasp) for a two-fingered gripper. [2]



(a) (b) (c)

Fig. 2. (a) Input point cloud; (b) grasp candidates that denote potential grasp configurations; (c) high scoring grasps.

Our grasp detection algorithm proceeds in two main steps. First, given an input point cloud (Figure 2 (a)), a set of 6-DOF grasp candidates are sampled using basic geometric constraints (Figure 2 (b)). These grasp candidates constitute a proposal distribution analogous to the object detection context. Second, a trained classifier predicts a binary grasp/not grasp label for each of these samples and assigns a probability with this prediction (Figure 2 (c)). We refer to this probability as the classifier's *confidence score*.

The classifier that predicts grasps needs to be trained in a supervised fashion with a labeled dataset. The ground truth labels in the dataset say which grasps are *true positives* and which are *true negatives*. *Accuracy*, which is the number of true positives and true negatives divided by the total number of predictions, quantifies the performance of the classifier. The labels can be generated using any reasonable grasp metric, human annotation, simulation (as in [10]), or the robot itself (as in [22]). In our system, we generate training data using BigBIRD [23], a dataset comprised of 125 3D object mesh models paired with 600 RGBD images taken from a hemisphere of different perspectives around the object. From the images we generate a large number of grasp candidates that serve as training examples. The ground truth for each candidate is calculated by using the 3D mesh model to evaluate whether a force closure grasp would result if the fingers were to close in that configuration.

III. Problem Statement

We assume the robot already knows approximately how it would like to pick up the object. For example, suppose the robot has detected a coffee mug that is to be picked up. The robot knows approximately how it should grasp the mug, but we would like to detect true positive grasps nearby the desired grasp with high confidence. More precisely,

Problem III.1 (Viewpoint Selection for Grasp Detection)**. Given** 1) a region of desired grasps (the center of which is called the *target grasp*), 2) the geometric category of the object in the vicinity of the grasp region (e.g. box-like,

[1] $SE(3)$ is the 6-DOF space of rigid body transformations.

[2] In fact, it is possible to use any grasp quality metric without changing the algorithm at all. The important thing is that there is a consistent, well-defined definition of a grasp.

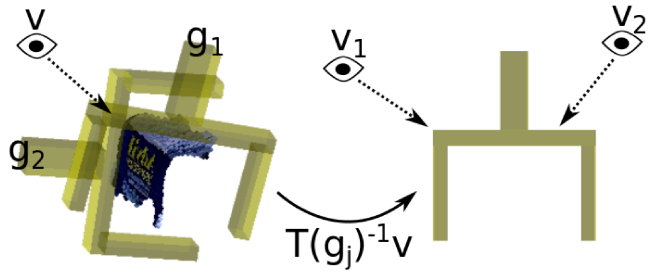


Fig. 3. Illustration of viewpoint projection into the reference frames of the grasps.

cylinder-like), 3) a desired number of grasps to detect, n, and 4) a set of potential viewpoints; **Calculate** the viewpoint from which the target grasp should be viewed in order to maximize the average accuracy over the top n scoring grasps.

In this paper, we assume the *region of desired grasps* is expressed in terms of a single target grasp pose and a neighborhood of positions and orientations about this desired pose. The objective is to maximize accuracy over the top n scoring grasps because, on a real robot, we typically need several grasps to choose from in order to satisfy kinematic and task constraints.

IV. Mapping Detection Performance as a Function of Viewpoint

In order to create an algorithm for selecting viewpoint, our first step is to characterize variables of interest (such as density of detected grasps, accuracy, etc.) as a function of viewpoint. Specifically, we create maps of the relevant variables where directions are azimuth (θ) and elevation (ϕ) of viewpoint relative to the target grasp. The map is created entirely using data from the BigBIRD dataset [23], comprised of real RGBD data for a variety of objects taken under controlled conditions. Each object in BigBIRD is associated with 600 different point clouds taken from different views in a hemisphere around the object.

A. Projecting viewpoint into the reference frame of a grasp

Since our maps will be expressed relative to the pose of the target grasp, we need to transform viewpoint into the grasp reference frame. We do this as follows. First, we run grasp detection on point clouds taken from a set of n different viewpoints $v_1, \ldots, v_n \in \mathbb{R}^3$, expressed as positions in a world reference frame (we assume that the camera is always pointed directly toward the target grasp). For each viewpoint $i \in (1, n)$, let $g_1^i, \ldots, g_{m_i}^i$ denote the set of m_i grasps detected from that viewpoint. Then, project each viewpoint into the reference frames of the grasps seen from that viewpoint. Figure 3 illustrates this idea. The left side shows two grasps that are detected in a single point cloud generated by viewing the object from v. When projected into the reference frames of the two grasps, we get the two viewpoints seen on the right side. Let $T(g)$ denote the 4×4 homogeneous transform that expresses the reference frame of grasp g relative to the world reference frame. Then, viewpoint i expressed in the reference frame of grasp j is $v_j^i = T(g_j^i)^{-1} v_i$.

B. Creating the maps

We create a map as follows. First, we detect grasps from many different viewpoints around several different objects. For each object, we sample 80 different viewpoints randomly from the 600 total views per object in BigBIRD. Then, we perform the viewpoint projection described above onto the grasps. The result is a set of viewpoints, $\mathbb{V} = (v_1^1, \ldots, v_{m_n}^n)$, expressed in the reference frame of the detected grasp. For each viewpoint, $v \in \mathbb{V}$, we store the confidence score assigned by the grasp detector to the corresponding grasp, $s(v) \in \mathbb{R}$, and the ground truth evaluation of whether the grasp was actually a force closure grasp or not, $gt(v) \in \{0, 1\}$.

BigBIRD consists of 125 objects that can be divided into roughly two shape classes: box-like objects and cylinder-like objects. Since these two object classes appear to have qualitatively different maps, we will characterize them separately. Figure 4 (a-e) shows the results of averaging the maps created for a set of 25 box-like objects from BigBIRD. Here, viewpoint is expressed in terms of the azimuth (θ) and elevation (ϕ) of the unit vector pointing toward the viewpoint expressed in the reference frame of the grasp. (Coordinates illustrated in Figure 1 (b) and Figure 5.) The maps are smoothed by convolving the data with an isotropic Gaussian with variance equal to 0.2. The convolved map values are computed at discrete positions spaced 0.05 radians (3°) and going out to ± 1.05 radians ($\pm 60°$) on both axes. Figure 4 (a-e) shows, respectively: the density of all detected grasps, the density of true positives (number of correct grasp detections), the density of false positives, grasp classification accuracy, and the density of true positives minus the density of false positives. Figure 4 (f-j) shows the same set of five maps, averaged over 14 cylindrical objects from BigBIRD. Notice whereas box-like objects are best viewed from a viewpoint of approximately $(\phi, \theta) = (0 \pm 0.1, 2.8 \pm 0.1)$, cylinder-like objects are best viewed from $(\phi, \theta) = (0.2 \pm 0.1, 2.9 \pm 0.2)$. Another interesting observation is that the grasp detection algorithm itself also makes a difference. The last row of Figure 4 shows the same statistics for a variant of grasp detection similar to the detector proposed by Kappler et al. [7] for the same set of 25 box-like objects. Whereas our default method samples grasp candidates by searching orientations

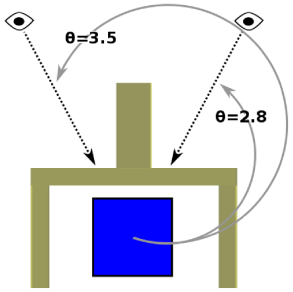


Fig. 5. Elevation angle of viewpoints from which to view a target grasp on a box-like object in order to maximize number of true positive minus false positive grasps detected.

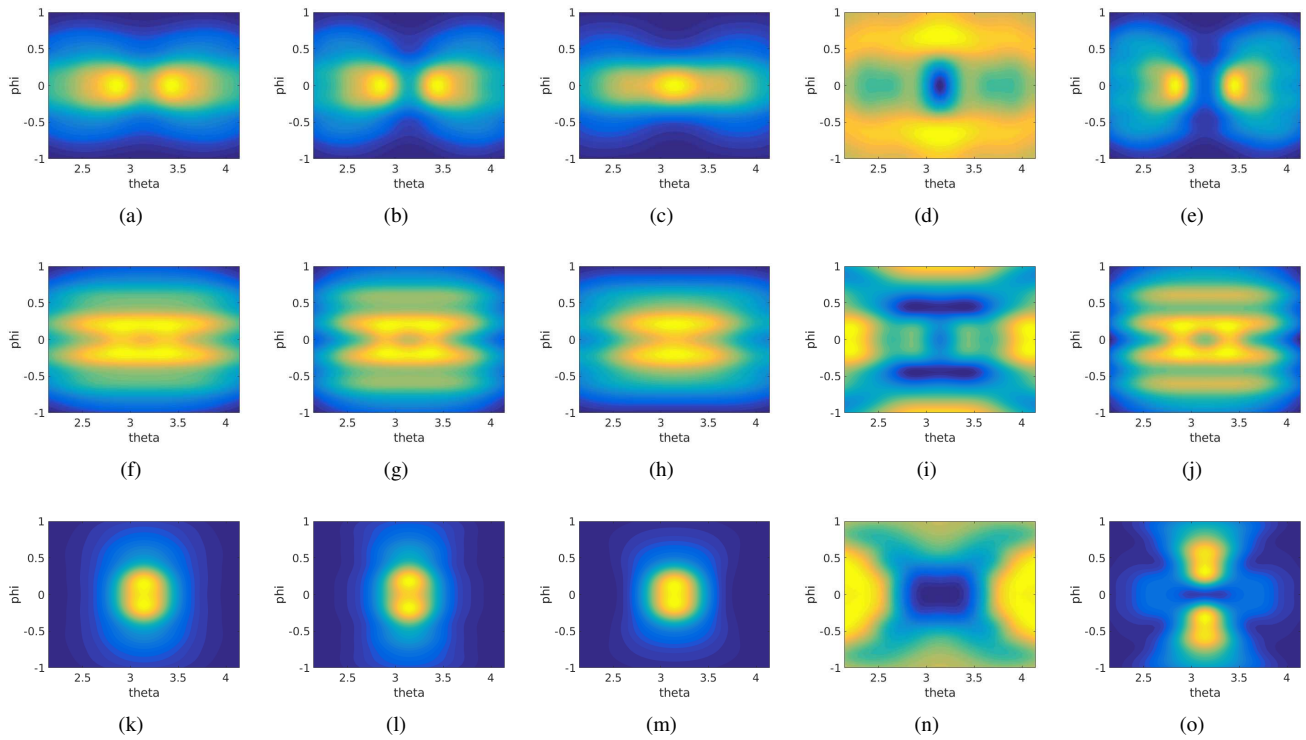


Fig. 4. Various performance metrics plotted as a function of viewpoint (azimuth and elevation). *Rows:* Top: results averaged over 25 box-like objects using default algorithm. Middle: results averaged over 14 cylinder-like objects using default algorithm. Bottom: results averaged over 25 box-like objects using Kappler algorithm variant [7]. *Columns:* First: density of all considered grasp candidates. Second: density of true positives. Third: density of false positives. Fourth: accuracy. Fifth: density of true positives minus density of false positives (our proposed measure).

about a local curvature axis, the method of Kappler et al. samples grasp candidates by searching orientations about a local normal axis. For the sake of comparison, we hold the method of classification fixed. Notice the shape of the map is completely different. This suggests the grasp detection algorithm also has an impact on viewpoint.

V. Viewpoint Selection

A. Approach

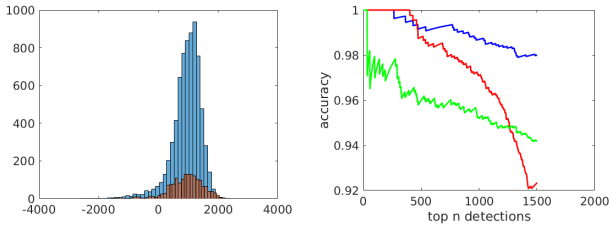
The key question is how the maps developed in the last section should be used to guide viewpoint. Perhaps the most obvious answer is to use classification accuracy, i.e. the maps shown in Figure 4 (d, i, n). However, there are a couple of problems with this choice. First, for all three object/algorithm variations, classification accuracy seems to be maximized at viewpoints associated with low-density grasp detection. Generating a large number of detections is important because it gives us a larger number of high scoring grasps to choose from, which will be crucial for a robot with kinematic and task constraints. Second, we do not empirically observe a significant improvement in accuracy in our offline evaluations by selecting viewpoint based on accuracy. It could be the expected improvement in accuracy is not large enough to make a difference, and it could be there is wider variation among objects as to which viewpoint maximizes detection accuracy.

Instead, we propose selecting the viewpoint that maximizes the expected number of true positives detected minus the expected number of false positives. This measure is shown in the right-most column of Figure 4. The idea is we want to maximize the number of true positives and minimize the number of false positives. An algorithm that selects viewpoints that maximizes this measure should generate a large number of true positives (i.e. a large number of correctly identified grasps) and thereby increase accuracy among the highest scoring grasps.

B. Offline experiments

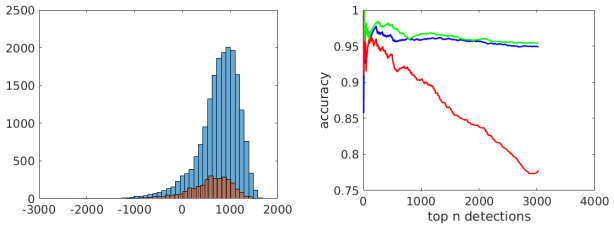
We ran experiments to compare grasp detection performance using 1) viewpoints chosen using the true positives minus false positives density map (*smart*), 2) viewpoints selected uniformly randomly from the set of possible viewpoints (*random*), and 3) head-on views of the target grasp (*head-on*). In the head-on contingency, we select from the set of possible viewpoints the one closest to a view that would see the object along the approach vector of the grasp. We performed this comparison offline using the point clouds contained within the BigBIRD dataset as follows. First, for a given object, select a ground truth grasp at random (using the mesh to evaluate ground truth). This will be the target grasp. [3] Second, for this target grasp, select a viewpoint using the proposed method, the head-on method, or at random out of the 600 views available for each BigBIRD object. Third, run grasp detection for the point cloud obtained from

[3]In practice, the target grasp would be selected based on task or object characteristics, but here it is selected randomly.



(a) (b)

Fig. 6. Grasp detection performance averaged over 25 box-like objects in BigBIRD: (a) histogram over the scores of the detected grasps using smart viewpoints (blue) and random viewpoints (red); (b) detection accuracy averaged over the top n scoring detected grasps. Blue: smart viewpoints. Green: head-on viewpoints. Red: random viewpoints.



(a) (b)

Fig. 7. Grasp detection performance averaged over 14 cylinder-like objects in BigBIRD: (a) histogram over the scores of the detected grasps using smart viewpoints (blue) and random viewpoints (red); (b) detection accuracy averaged over the top n scoring detections. Blue: smart viewpoints. Green: head-on viewpoints. Red: random viewpoints.

that view and prune out all grasps that are not within the neighborhood of the desired grasp. We evaluate the number of detected grasps and the detection accuracy of the top n scoring detections averaged over all objects within a category.

	True Positives	Positives	Accuracy
Boxes Smart	5448	6123	0.89
Boxes Head-on	1923	2206	0.87
Boxes Random	1312	1424	0.92
Cylinders Smart	13834	17150	0.80
Cylinders Head-on	14291	18112	0.79
Cylinders Random	2155	2754	0.78

TABLE I

SUMMARY OF VIEWPOINT SELECTION RESULTS. COUNTS ARE TOTALED FROM ALL VIEWPOINTS AND OBJECTS, AND ACCURACY IS AVERAGED OVER ALL VIEWPOINTS AND OBJECTS.

Figure 6 shows the results averaged over the 25 box-like objects in BigBIRD. Figure 6 (a) shows a histogram over scores for all positives detected. The larger blue histogram shows the scores of the grasps detected using smart viewpoints. The smaller red histogram shows the same for random viewpoints. Figure 6 (b) shows accuracy for the top n scoring grasps using the three viewpoint selection methods. In each contingency, we detected a large number of grasps over all objects within the category (either box-like or cylinder-like) and pruned those that were not within the desired neighborhood of the target grasp (2 cm and 20°) in our experiments). Of the remainder, we ranked them by the score produced by the deep network used to classify the grasps (higher means more "confident") and evaluated classification accuracy over the top n scoring grasps. Out of the three methods, the head-on view (the green line in Figure 6 (b)) is clearly at a disadvantage. The reason can be seen by looking at the maps for true positives and accuracy (Figure 4 (c, d)). The head-on view "sees" the target grasp from $\phi = 0, \theta = 0$. However, this view is associated with low classification accuracy and high numbers of false positives. Also, notice that the top-n accuracy for random views (the red line in Figure 6 (b)) drops off significantly for larger values of n. Even though random does happen to detect some good grasps, it does not detect as many of them as does the proposed method (the blue line in Figure 6 (b)).

Figure 7 shows similar results for the group of 14 cylindrical objects in BigBIRD. Figure 7 (a) shows that the random viewpoint method detects approximately 6 times fewer positive grasps than does our proposed method. Figure 7 (b) shows that top-n classification accuracy for random viewpoints (the red line) drops off quickly, as it did for box-like objects. Interestingly, the head-on method performs similarly to the proposed method for cylinder-like objects. This is because the head-on view for cylinders turns out to be a relatively good view (see $\phi = 0, \theta = 0$ in Figure 4 (g, i)). For cylinder-like objects, the proposed method selects views that nearly approximate the head-on view.

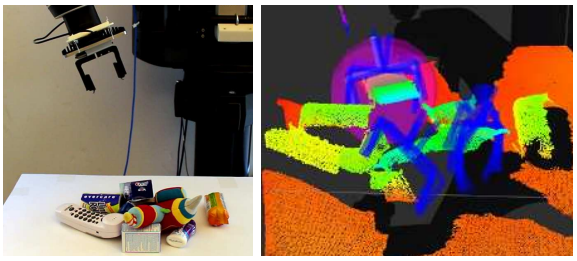
VI. ROBOT EXPERIMENTS

The results reported so far indicate that our proposed viewpoint selection method can improve the accuracy with which the top n grasps are detected. But, how well does this translate into grasp success on a real robotic system? In this section, we evaluate the approach in the context of a robot grasping in dense clutter.

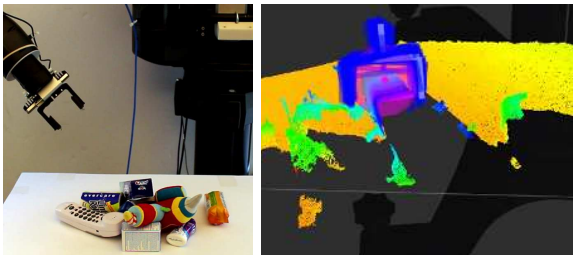
A. Setup

In these experiments, each grasp proceeds as follows. First, we obtain a target grasp by taking a view of the objects from a random viewpoint and running grasp detection on this point cloud. We select one of the detected grasps based on confidence score and task-specific heuristics (such as height in the pile and how close to vertical the approach vector is). Then, we use the smart viewpoint selection strategy to obtain a viewpoint from which to view the target grasp (we select the best viewpoint subject to inverse kinematics constraints). Next, we detect grasps in this new point cloud. Since we already know the approximate location of the target grasp, we speed up the second round of grasp detection by only searching a small region (8 cm radius ball) about the grasp target. Finally, we select the highest scoring grasp (again, subject to IK constraints) and execute it. This process is illustrated in Figure 8 (a-b). We measure success in terms of how often the grasp succeeds.

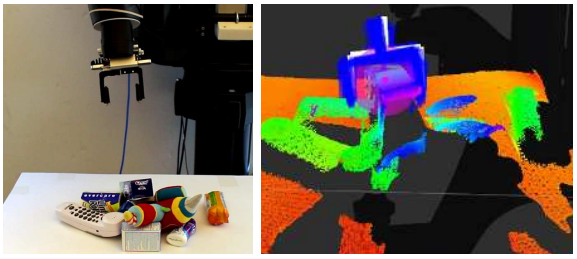
In principle, our proposed viewpoint selection method should use the grasp density map corresponding to the shape of the object to be grasped. However, since our scenario



(a) First view (random)



(b) Second view (smart)



(c) Third view (alignment)

Fig. 8. Illustration of the three-view grasp detection strategy. The point cloud and grasps (right) are obtained from the depth sensor (left). The white grasp is the target grasp used for planning the next step.

involves a variety of objects piled together, we just used a single map for all grasp attempts. Since the viewpoints that maximize our proposed viewpoint quality measure (density of true positives minus density of false positives) for box-like objects (Figure 4 (e)) also nearly maximizes the measure for cylinder-like objects (Figure 4 (j)), we used the map for box-like objects all the time.

In addition to the smart and random viewpoints, we also evaluated performance for a third viewpoint taken directly in front of the target grasp ($\phi = 0, \theta = 0$ in Figure 1 (b)). We moved the sensor as close to the target grasp as possible while remaining outside the minimum viewing depth for the sensor (20 cm in our case). Instead of running the full grasp detection algorithm on the view obtained from this perspective, we just ran the candidate generation part of the algorithm and accepted the candidate most closely aligned with the target grasp. We call this the *alignment* view. Its purpose is to help correct for kinematic errors in the robot: the effect of the robot's kinematic errors is limited to only those errors that accumulate while the arm travels from the the view pose to the grasp target. In order to isolate the effects of the various different views, we performed experiments for all relevant variations on viewing order: 1-2-3 (random, smart, alignment), 1-2 (random, smart), and 1-3 (random, alignment).

(a) (b)

Fig. 9. (a) All 25 objects used in robot experiments. (b) Cluttered pile of 10 objects that the robot must clear.

The experimental protocol followed for each variation is similar to the one proposed in our prior work [9]. First, 10 objects are selected at random from a set of 25 and "poured" into a pile in front of the robot. (See Figure 9 (a) for the object set and Figure 9 (b) for an example pile of clutter.) Second, the robot proceeds to automatically remove the objects one-by-one as the experimenter records successes and failures. This continues until either all of the objects have been removed, the same failure occurs on the same object three times in a row, or no grasps were found after three attempts. The sensor (Intel RealSense SR300) and gripper hardware used in the experiment are shown in Figure 1 (a). Figure 10 shows the robot performing the first five grasps of one round of an experiment. [4]

B. Results

	Views 1-2-3	Views 1-2	Views 1-3	Views 1
Attempts	131	141	154	153
Failures	17	31	26	39
Success Rate	0.87	0.78	0.83	0.75

TABLE II

GRASP SUCCESS RATES FOR THE FOUR EXPERIMENTAL STRATEGIES.

Grasp success rates for the four experimental strategies are shown in Table II. The "attempts" row of the table shows the number of grasp attempts made using each strategy, the "failures" row shows the number of failures for each strategy, and the "success rate" row is one minus the ratio between failures and attempts. The grasp failures in our experiments primarily fell into two categories as shown in Table III. The "FK" row in Table III denotes the number of grasp failures caused by a difference between the planned grasp and the actual grasp in the real world. The "grasp" row denotes the number of grasp failures caused by a detection error. The failure modes seem to reinforce intuition about what should happen if either view 2 (smart) or view 3 (alignment) is skipped. If the alignment view is skipped, then we obtain a large number of FK failures, presumably because we are not registering the point cloud close to the final grasp configuration. If the smart view is skipped, then we obtain relatively more detection failures because the algorithm does

[4]A video illustrating the experiment is available at `https://youtu.be/iGRbqFsNgzo`.

Fig. 10. First 5 grasps of a typical experiment trial. All grasps were successful. This was a run with all 3 views included.

not get the best view of the target grasp (in terms of detector performance).

Perhaps the most noticeable result from this experiment is that adding the third view helps: going from 1-2 to 1-2-3 adds 9% to the grasp success rate; going from 1 to 1-3 adds 8%. However, this is increase (and the relatively poor performance of grasping without the alignment view) is a result of kinematic errors in the Baxter robot and/or calibration errors in the Intel RealSense SR300 sensor we used. Adding the alignment view helped to correct for these errors. However, the benefits of adding the alignment view should not overshadow the additional benefit of the smart view. Without the alignment view, adding the smart view increases the grasp success rate from 0.75 to 0.78 (a 3% increase). With the alignment view, adding the smart view increases grasp success from 0.83 to 0.87 (a 5% increase).

(Error Type)	Views 1-2-3	Views 1-2	Views 1-3	Views 1
FK	10	21	9	28
Grasp	4	10	14	9
Other	3	0	3	2

TABLE III

COUNTS BY FAILURE TYPE FOR THE FOUR EXPERIMENTAL STRATEGIES. "FK" MEANS A FORWARD KINEMATICS ERROR LED TO THE FAILURE AND "GRASP" MEANS A DEFECT IN THE DETECTED GRASP LED TO A FAILURE.

VII. CONCLUSION

Our main conclusion is viewpoint can have a significant effect on the performance of grasp detection. The right viewpoint can enable grasp detection to find 4-6 times the number of good grasps relative to an uninformed view. Our results show this increase in the number of detected grasps can have a significant effect on the average accuracy of the top detected grasps. These results are borne out in robotic experiments on our Baxter showing an improvement in overall grasp success rates using an informed viewpoint selection method.

REFERENCES

[1] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The Int'l Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[2] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The Int'l Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[3] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *IEEE Int'l Conf. on Robotics and Automation*, 2015, pp. 1316–1322.

[4] D. Fischinger, M. Vincze, and Y. Jiang, "Learning grasps for unknown objects in cluttered scenes," in *IEEE Int'l Conf. on Robotics and Automation*, 2013, pp. 609–616.

[5] R. Detry, C. H. Ek, M. Madry, and D. Kragic, "Learning a dictionary of prototypical grasp-predicting parts from grasping experience." in *IEEE Int'l Conf. on Robotics and Automation*, 2013, pp. 601–608.

[6] 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, pp. 2379–2384.

[7] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *IEEE Int'l Conf. on Robotics and Automation*, 2015.

[8] A. ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," in *Int'l Symp. on Robotics Research*, 2015.

[9] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2016.

[10] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*. IEEE, 2016, pp. 4461–4468.

[11] D. Kanoulas, J. Lee, D. G. Caldwell, and N. G. Tsagarakis, "Visual grasp affordance localization in point clouds using curved contact patches," *Int'l Journal of Humanoid Robotics*, 2016.

[12] O. Kahler, P. V. Adrian, R. C. Y. Yuheng, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile device," in *IEEE Int'l Symp. on Mixed and Augmented Reality*, vol. 22, 2015, pp. 1241–1250.

[13] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The Int'l Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[14] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: a survey," *Pattern Recognition*, vol. 37, no. 3, pp. 429–446, 2004.

[15] J. Velez, G. Hemann, A. S. Huang, I. Posner, and N. Roy, "Planning to perceive: Exploiting mobility for robust object detection," in *Int'l Conf. on Automated Planning and Scheduling*, 2011.

[16] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy," in *Int'l Conf. on Vision Modeling and Visualization*, vol. 1, 2001, pp. 273–280.

[17] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.

[18] H. Van Hoof, O. Kroemer, H. B. Amor, and J. Peters, "Maximally informative interaction learning for scene exploration," in *IEEE Int'l Conf. on Intelligent Robots and Systems*, 2012, pp. 5152–5158.

[19] D. Levine, B. Luders, and J. P. How, "Information-rich path planning with general constraints using rapidly-exploring random trees," in *AIAA Infotech@ Aerospace Conf.*, 2010.

[20] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[21] G. Kahn, P. Sujan, S. Patil, S. Bopardikar, J. Ryde, K. Goldberg, and P. Abbeel, "Active exploration using trajectory optimization for robotic grasping in the presence of occlusions," in *IEEE Int'l Conf. on Robotics and Automation*, 2015, pp. 4783–4790.

[22] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The Int'l Journal of Robotics Research*, 2016.

[23] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird:

A large-scale 3d database of object instances," in *IEEE Int'l Conf. on Robotics and Automation*, 2014, pp. 509–516.