Abstract—Intelligent object manipulation is critical for a robot to effectively operate in a household environment. There are many grasp planners that can estimate grasps based on object shape, but these approaches often perform poorly because they miss key information about non-visual object characteristics. Object model databases can account for this information, but existing methods for database construction are time and resource intensive. We present an easy-to-use system for constructing a grasp database from crowdsourced demonstrations. The method requires no additional equipment other than the robot itself, and non-expert users can demonstrate grasps through an intuitive web interface, with virtually no training required. We show that the crowdsourced grasps can prove sufficient for object manipulation, and furthermore the demonstration approach outperforms purely vision-based grasp planning approaches for a wide variety of object classes.

I. INTRODUCTION

Intelligent object manipulation is a critical skill for robots to operate autonomously in household environments. Robots often need to interact with a large variety of objects within these environments, and as such they must either make use of a database of known objects, or automatically determine useful properties of any objects encountered. The latter approach can lead to difficulties in determining usability characteristics for arbitrary objects, e.g. a bottle of water should not be grasped in such a way that the opening is obstructed, preventing pouring.

It is difficult to determine usability information for a set of objects using an autonomous process without any prior knowledge of the object, as environments can have a large number of objects that must all be grasped with different criteria in mind. Some usability information can be generalized, such as the pouring metric mentioned above, but this metric only applies to certain sets of objects. Complicating the grasping issue further, some difficult-to-observe physical characteristics of objects, such as weight distribution or fragility of materials, need to be accounted for when performing a successful grasp.

Grasp demonstration by human users provides a natural way to convey hard-to-observe information to robots. Demonstrating grasps for an object set can be repetitive and time consuming, however. Web robotics allows for the collection of demonstrated grasps from a variety of users. Furthermore, crowdsourcing helps to reduce the time and effort required from each individual user while still providing large amounts of data. We present a method of incorporating data collected from crowdsourcing into an object recognition and grasping database, resulting in an automated grasping technique that inherently accounts for usability constraints.

This paper builds on our previous work in crowdsourced object recognition database construction [1], in which we conducted a user study to collect object recognition and grasping data, and presented a system to construct an object recognition database using that data. Here, we extend our previous work to include grasping data, while still meeting the original goals of the recognition database, i.e. that it should both be easy to add new objects and require no equipment beyond the robot and the objects themselves. Both this ease of use and lack of setup allows the process to be crowdsourced, which is appealing to researchers who want to quickly expand the capabilities of their system. We also compare the effectiveness of the crowdsourced grasps to expert demonstrated grasps, and we show the advantages that this grasping system has over a grasp planner based solely on the geometry of the object.

II. RELATED WORK

Many methods exist for vision-based grasp planners using depth image data. These approaches often forgo object recognition and rely solely on geometric analysis of partial views of novel objects to compute effective grasp points [2], [3], [4]. Vision-based grasp planners can be effective for grasping novel objects, but they do not account for object information beyond visible shape.

In this work we use Willow Garage’s PR2 robot. This platform has two common off-the-shelf grasp planners. One uses depth data with no object recognition component [5], which is susceptible to the same problems as partial shape grasp planners. The other grasp planner incorporates Willow Garage’s household objects database [6], which matches detected objects to 3D models, and includes a set of example grasps calculated for each model using the GraspIt! simulator. The database was designed to work with a limited set of commonly available objects. It relies on detailed 3D modeling and is constrained to objects that are rotationally symmetric or have no indentations or concavities. The database is difficult to use in new environments because of the difficulty in adding new object models. Additionally, the grasps in the household objects database were calculated solely based on object shape and do not account for other object properties. Our method aims to solve the same problem as the PR2’s
grasp planners, but we seek to improve the grasps’ quality from a usability standpoint, remove the restrictions on object shape, and simplify the process of adding new objects.

Work has also been done on autonomous grasp evaluation based on usability characteristics. Baier and Zhang [7] present a set of grasp usability criteria, and include analytical methods for evaluating them. Some criteria are specific only to kitchen objects, although others generalize well to all objects. Baier and Zhang are successful in representing a set of common usability characteristics, but there are many unrepresented characteristics specific to other objects. Additionally, there is also the issue of determining which criteria to apply to which objects. Both of these issues make it difficult to explicitly define usability characteristics for arbitrary object sets.

These usability characteristics are, however, naturally known by humans, and Xue et al. present a system for leveraging human knowledge and non-vision based object characteristics into a multi-modal grasp planning system [8]. The system requires data collection of an object’s shape, texture, and weight, and allows human instructors to demonstrate areas to be touched or avoided during grasping. The drawback of this approach is that it requires a complicated setup, including a digitizer, turntable, movable stereo camera setup, and tactile glove. This approach is also unsuitable for crowdsourcing, as it requires a trained demonstrator to give semantic information about each object.

Other object model databases exist that include grasping information, such as the partial-view-based data-driven grasping system presented in [9]. This system adds the object recognition component to the partial-view depth image grasp planners, and retrieves grasp points from an online database. The system is more concerned with the recognition aspect, though, and does not address where the grasp data comes from. Another object model and grasping database addresses the problem of defining grasp points by using a set of grasps exhaustively calculated offline from CAD models [10]; this requires any object to be modeled in detail before it can be used. Similarly, Goldfeder et al. present a large-scale database constructed with grasps calculated using the GraspIt! simulator [11], which replaced human grasp demonstrations, reducing time constraints in constructing a large-scale database. As a result, the database includes many baseline form closure grasps for each object.

An alternative to calculating grasp points is to learn them through trial and error. Detry et al. present an autonomous experimentation system where a robot repeatedly attempts object grasps at various points and adjusts probabilistic grasping models accordingly [12]. We use a similar idea for evaluating grasp models, except we use crowdsourced human-demonstrated grasps to decrease the time required by the trial-and-error learning system, while also leveraging human knowledge of object usability characteristics.

Applying crowdsourcing methods to grasp learning is not new. Sorokin et al. used Amazon’s Mechanical Turk (AMT) to crowdsource novel object grasping by dividing object modeling into subproblems of object labeling, human-provided segmentation, and final model verification [13]. They demonstrated successful object recognition and grasping using the results obtained from AMT workers. Microtask crowdsourcing marketplaces do have some limitations, however. The system does not allow for direct control of a robot, and researchers must design tasks to be simple and quick to ensure for high quality data. With new web robotics systems such as the Robot Management System (RMS) [14], we can perform more complicated tasks, including the grasp pose demonstrations used in this paper.

III. GRASP LEARNING

In this section, we present a complete grasp learning system. We first briefly discuss the data collection process, followed by a description of how grasps are mapped to the 3D object models. We then present an outlier filtering algorithm that removes clearly erroneous grasps, and we conclude with an online training algorithm that learns probabilistic success rates for the remaining grasps, to both determine the order in which grasps should be attempted and remove any erroneous grasps missed during the outlier filtering phase. The result is a database of object models with associated grasp poses ordered by success rate. Figure 3 provides examples of visualized output from each major step of the process.

A. Data Collection

We designed the grasp demonstration collection phase to work with a minimal amount of setup, so that new grasps and object models can be added to the database with ease. As such, the data collection requires only a robot with a depth sensor that can generate point clouds, such as a Microsoft Kinect. Through teleoperation, a user can demonstrate a grasp pose for a desired object, and the data collection system will then store a segmented point cloud representation of the object, the demonstrated grasp pose, and a user-provided object label.

Both expert and non-expert users can demonstrate grasps for this system. In this paper, we present the results of grasping data from non-expert users, collected in a crowdsourcing user study. The study consisted of remote users connecting to a PR2 over the Internet, conducted using RobotsForMe [14]. We instructed participants to demonstrate as many grasps on as many objects as they could within a 20 minute time limit by teleoperating the robot using a web interface. The study used nine objects, shown in Figure 1. Further details of the user study can be found in our previous work [1]. Additionally, we compare results against grasps demonstrated by an expert.
B. Model Construction

After data collection, the next phase of the grasp learning pipeline combines the object data into 3D object models with associated grasp poses for each model. Our object model construction algorithm conducts point cloud registration to combine each view of an object into a single model [1]. This algorithm uses a set of metrics to evaluate point cloud registration, which are used to intelligently determine the order in which to perform successive pairwise point cloud registrations based on a graph structure. The pairwise registration pipeline itself uses functionality from the Point Cloud Library (PCL) [15].

In this paper, we extend the registration algorithm to combine the individual grasps into a list of grasps associated with the object model. Each pairwise registration within the algorithm produces a transformation matrix. Applying this transformation matrix to the demonstrated grasp pose at each pairwise registration step results in each object model also containing an associated set of grasps in its local reference frame. Figure 2 shows one of the object models generated from the user study, and further examples are shown in the left column of Figure 3.

C. Outlier Filtering

With the object models constructed, the next step of the pipeline focuses on evaluating the usefulness of the demonstrated grasps. This is a crucial step because demonstrated grasps can result in low quality grasp poses, due to poor object segmentation or grasps that accidentally nudge an object before lifting it up. This is particularly true when crowdsourcing grasps demonstrated by non-expert users, where quality of data cannot be guaranteed.

The grasp training algorithm, described further in the next section, also detects and removes unsuccessful grasps from an object model. This is an online algorithm that requires the robot to determine each grasp’s quality through trial and error, and as such it becomes time consuming with large numbers of grasps. The goal of the outlier filtering phase is to eliminate as many grasps as possible with an offline algorithm, before the object models are passed through to the online grasp training phase.

With only a point cloud representation of an arbitrary object, it is difficult to determine the effectiveness of a grasp based solely on its location relative to the point cloud. To prevent the removal of potentially successful grasps, the only grasps that can be removed with high certainty are those that lie on the outside of the point clouds. Specifically, the algorithm fits a bounding box around the point cloud, and removes any grasps that are located outside of the bounding box at a distance greater than half the length of the robot’s open gripper. These grasps will most likely miss the object entirely; this can be seen in Figure 3, where one outlying grasp is removed for both the bowl and the phone.

D. Grasp Training

The final phase of the grasp learning pipeline learns probabilistic success rates for each grasp, which are then used to determine a grasp order for each object. An epsilon-greedy algorithm, Algorithm 1, learns these probabilities by testing each grasp repeatedly with the robot. The algorithm begins with a list of grasps to be tested, an empty list for storing sufficiently explored grasps, and by initializing the exploration variable, $\epsilon$ (lines 1-2). The algorithm continues to select and test grasps until every grasp associated with a model has been attempted a minimum of $N$ times (lines 3 and 13). During each loop iteration, the algorithm selects grasps by either randomly exploring the set of untested grasps, or by selecting an already explored grasp with the highest chance of success (lines 4-9). The robot then performs the selected grasp, evaluates its success, and updates the success rate and number of grasp attempts accordingly (lines 10-12).

The value of $\epsilon$ determines the exploration strategy by which new or previously attempted grasps are selected. Higher values of $\epsilon$ result in more frequent exploration; conversely, lower values of $\epsilon$ result in more frequent execution of the previously explored grasps. The algorithm begins with a high value of epsilon, which decays exponentially (line 17).
Algorithm 1 Epsilon-greedy grasp training

Require: List<Grasps> grasps
1: List<Grasps> exploredGrasps;
2: $\epsilon = 1$;
3: while grasps.size() > 0 do
4: $r = \text{rand}(0, 1)$;
5: if $r < \epsilon$ then
6: Grasp $g = \text{pickRandom}(\text{grasps})$;
7: else
8: Grasp $g = \text{maxGraspProbability}(\text{exploredGrasps})$;
9: end if
10: Boolean success = testGrasp(g);
11: updateGraspProbability(g, success);
12: $g$.numAttempts ++;
13: if $g$.numAttempts $\geq N$ then
14: $\text{exploredGrasps}$.push(g);
15: $\text{grasps}$.remove(g);
16: end if
17: $\epsilon = 0.975 \times \epsilon$;
18: end while

as the training continues. As such, the algorithm first spends most of its time attempting unexplored grasps, and as time goes on, it spends more time refining the success rates for the learned grasps. Over time, $\epsilon$ decays to zero, and further exploration of untested grasps will no longer occur. For larger grasp sets, such as a large crowdsourced dataset of grasp demonstrations, the algorithm can be adjusted to terminate after it finds a sufficient number of successful grasps, by changing the loop condition in line 3.

Upon completion of the training, the algorithm discards any grasps with a success rate of zero from the model. For example, Figure 3 shows one grasp removed from both the bone and the phone after the training algorithm determined that the grasps had no chance of success. The example also shows high-probability ($\text{graspProbability} > 0.5$) grasps in green. These grasps will be attempted first, and the low-probability grasps ($0 < \text{graspProbability} \leq 0.5$), shown in red, will only be attempted if all of the high-probability grasps are unreachable from the current robot position.

The results presented below used training with $N$ set to 3. Increasing $N$ can increase the accuracy of the success rates, at the trade-off of increasing training time. To make up for this short training time, we continue to update the grasp model online after the dedicated training process is complete. This lifelong learning could lead to slow adaptability once the number of grasp attempts gets high, which could be improved by keeping grasp success data over a sliding window of previous attempts. For this paper, we did not perform any long-term testing with the grasping database, and all results were gathered using this continual learning. We leave improvements for long-term use to future work.

IV. RESULTS

In the following section, we present an evaluation of the grasp learning system itself, comparisons of crowdsourced data to expert user demonstration, and a comparison of the demonstrated grasp database to a more traditional vision-based grasp planner. We also provide further examples of situations where the grasp learning system outperforms shape-based planners.

A. Evaluation of Crowdsourced Data

The user study provided us with a varying number of grasp demonstrations per object, shown in Table 1. Using the grasp learning pipeline presented above, we created a database of object models with associated grasps from the user study data. To evaluate this process, we performed an experiment in which the PR2 attempted to grasp the objects in randomized positions using the learned database for grasp planning. For a point of comparison, the researchers demonstrated grasps for the same object set, by using an offline equivalent of the web-based user study teleoperation interface implemented in rviz, a 3D visualization tool for the Robot Operating System (ROS). The researchers provided ten demonstration grasps for each object. Also, in order to evaluate the approach as a whole, we performed the same experiment using the PR2’s standard grasp planning algorithm [5] instead of our database. The results are shown in Figure 4.

In general, the PR2 grasped the objects with a success rate of at least 80% using grasps learned from both the user study data and the researcher demonstrated data. Of note is the black cup, which failed under all three grasping systems. This occurred because the Kinect could not properly segment the object due to its reflectiveness and the lighting conditions of the experiment. Another outlier is the truck, which was rarely grasped successfully during the user study, and the algorithm could not construct a complete object model from the crowdsourced data. The final outlier of the user study data is the dog bone, which was grasped much less successfully using the user study data. This was likely due to the fact that the user study data produced a comparatively low number of high-probability grasps for this object, as shown in Figure 5. We suspect that with more crowdsourced data, the bone

![Figure 4](image-url)
TABLE I

NUMBER OF SUCCESSFUL GRASPS DEMONSTRATED BY USER STUDY PARTICIPANTS.

<table>
<thead>
<tr>
<th>Object</th>
<th>Grasps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basket</td>
<td>22</td>
</tr>
<tr>
<td>Bone</td>
<td>15</td>
</tr>
<tr>
<td>Book</td>
<td>10</td>
</tr>
<tr>
<td>Bowl</td>
<td>27</td>
</tr>
<tr>
<td>Cup</td>
<td>3</td>
</tr>
<tr>
<td>Dragon</td>
<td>8</td>
</tr>
<tr>
<td>Duck</td>
<td>8</td>
</tr>
<tr>
<td>Phone</td>
<td>16</td>
</tr>
<tr>
<td>Truck</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 5. Comparison of the ratio of high-probability (graspProbability > 0.5) grasps to total grasps for the user study data and the researcher demonstrated data.

would be grasped as effectively with the user study data as with the researcher demonstrated data.

Comparing the demonstrated grasps to the grasps generated from the PR2’s planner, we can see some interesting results. For many of the objects, the methods have little difference in success rate, and no method clearly outperforms the others. In a few specific cases, however, the learned grasps outperform the geometrically planned grasps. The first case is the basket, which has many potential grasp areas due to its subdivided sections, as well as the handle on top. The geometric planner had difficulty determining which grasp would be most effective, and often selected grasps that were blocked by the handle or the inner edges of the subdivided sections. The demonstrated grasps, however, often picked up the object by the handle on top, since that is where a human would naturally grasp the object. This grasp was more successful with the PR2, since the handle is in an open area not blocked by other parts of the object. The second case is the duck, which was simply too small for the PR2’s planner to consistently analyze. The learned grasps worked well, though, since they required no planning. The final case is the truck, where the geometric planner occasionally failed due to grasping the moving wheels of the object, which caused the truck to slip from the gripper. Thanks to human knowledge of the truck’s moving wheels, the demonstrated grasps did not have this problem.

In each case, the learned grasps are superior to the geometrically calculated grasps for overall grasping success rate. Figure 6 shows a comparison for each object. For the water bottle and coffee creamer, switching to a grasp from the side at an appropriate height improved the grasp success rate to 100%; accounting for the balance point significantly improved the success rate for the hammer, which otherwise was grasped at random positions along the handle; for the monkey and the vase, the demonstrated grasps resulted in successes where the geometrically planned grasps could not succeed at all.

B. Advantages of Demonstrated Grasps

Following the results of comparing the demonstrated grasps to the PR2’s geometrically calculated grasps, the researchers demonstrated additional grasps for a supplemental object set. Each object in this set represents a special case of constraints for grasping, which the geometric planner was unable to detect. Figure 7 shows each supplemental object with both an example calculated grasp and an example demonstrated grasp. The first object, the water bottle, is difficult for the PR2 to grasp because most of the bottle’s surface is too slippery for the gripper to hold securely. There is a graspable region at the neck of the bottle, but the geometric planner would often grasp below it on the smoother surface. Next is the coffee creamer, which presents a challenge to the PR2 since its diameter is about the same distance as the width of the robot’s open gripper, causing small miscalculations to result in missed grasps. The grasp planner favors grasps from above, but the object has a much more consistent grasp from the side. The hammer represents objects with extreme weight distributions, which slip from the gripper when the object is not grasped near its balance point. The monkey has a removable part, its hat, and again due to the geometric planner’s preference for grasps from above, the PR2’s planner picks up only the hat instead of the monkey. Finally, the vase represents objects containing fragile parts. The PR2’s grasp planner always attempts to grasp the flower, which will both ruin the flower and fail to pick up the vase.

In each case, the learned grasps are superior to the geometrically calculated grasps for overall grasping success rate.
V. CONCLUSIONS

While purely vision-based approaches to grasp planning can prove sufficient for some objects, the inclusion of more information is necessary for a robot to grasp arbitrary objects with a high rate of success. For many objects, identifying key characteristics that determine the appropriate grasp location is beyond existing automated techniques. This problem can be circumvented by incorporating databases that already account for difficult to identify characteristics, which can be constructed from grasp demonstration.

In this paper, we showed that by using crowdsourcing, we can leverage human knowledge to create databases for autonomous object manipulation. We have shown that, with a sufficient amount of data collected and an appropriate system for determining which data is useful, non-expert users can demonstrate grasps that a robot can use to successfully manipulate household objects. Furthermore, this data can be collected using a minimal amount of setup, requiring nothing more than the robot and the object set. Due to the varied quality of crowdsourced data, this system requires more data collection than if expert users were demonstrating the grasps. The advantages of crowdsourcing mitigate this extra data collection, though, as it still requires less time and effort per individual user than if a researcher took the time to demonstrate all of the grasps on their own.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation award number 1149876, PI: Sonia Chernova, and ONR Grant N00014-08-1-0910, PI: Odest Chadwicke Jenkins.

REFERENCES