CS229 Final Report: Learning to Select a Good Grasp

Lawson L.S. Wong
STAIR - Perception-Manipulation

December 12, 2007

1 Introduction

Robotic grasping is a fundamental aspect of robotic manipulation, as the robot must first know how to acquire objects in any given and possibly unknown environment before it can work around in that place. Grasping is an essential skill for manipulators, and almost all more advanced manipulation tasks rely on this basic skill; once reliable grasping is in place, endless opportunities will become available. The general problem of grasping is, given a model of the environment (either known *a priori*, or in the context here, constructed from a camera point cloud), to find a grasping point on an object (possibly a given target) in the environment *and* to execute a successful grasp of the object at that point.

In the context of the Stanford AI Robot (STAIR), where the external environment is unknown, solving the described problem of grasping requires integrating techniques from perception (acquiring environment model from camera), grasp point identification (finding a grasp point on any object), and arm motion planning (to execute a collision-free path to the object). These three aspects have all been solved previously by the STAIR team [1, 2]. However, a simple pipeline consisting of just these three components, which is similar to the current system, is insufficient; even if each part works robustly and successfully produces a feasible motion of the arm to the found grasp point, a 'good' grasp cannot be guaranteed. Here we define a grasp, which is a particular robot hand configuration, as 'good' when upon closing the hand synchronously from its configuration, the target object (e.g., bowl) can be lifted up from its support surface (e.g., table) without it slipping out of the hand. The reason the simple pipeline is incapable of producing good grasps most of the time is that identifying feasible *grasp points* is different from identifying feasible *good grasps*. The goal of this project is to close this problematic gap.

The problem for this project is therefore, given the model of the environment and the planned hand configuration for grasping, to classify whether this hand configuration will result in a good grasp or not. More generally, given a set of such configurations, the objective is to determine the best hand configuration for the robotic arm to execute. To achieve this, a model of grasping was learned based on features that both intuitively and effectively distinguish good grasps from bad ones. This approach and its implementation on the STAIR platform will be discussed next.

2 Approach

The 3D point cloud of the environment is given by a SwissRanger laser scanner, which returns a grid of 144*176 points. As most of these points belong to large objects such as walls or tables and hence are irrelevant to the target object, it would be inefficient to consider such points that provide little information. Therefore, only a local region of the point cloud near the arm's hand will be used. For the Barrett Arm on STAIR 2, the target platform for this project, the *hand tip* shall be defined as 10cm in the 'out' direction from the hand's palm, which is approximately the length when the three fingers of the hand are at full extent outwards. The *local region* will then be defined to contain any point within 10cm of the hand tip. This means that the local region is roughly 'two hands' length' in each direction, which captures information about the area within the hand's grasp and its immediate surroundings.

An important advantage of using the local region is that only around 500 3D points will need to be processed when computing features, instead of the over 25,000 points returned from the camera. This shortens processing time to within several seconds. An additional advantage is that for most large objects such as plates and bowls, the edge at which the grasp points are found usually has a locally planar surface. Planar grasps are generally easier, so only considering the local region reduces the more complex problem of grasping arbitrarily shaped large objects to the simpler problem of grasping a planar surface on the object. Due to this advantage, and because the camera calibration is not yet precise enough to model small objects accurately, only large objects will be considered for this project.

Features of the grasp that can distinguish between good and bad grasps will be computed from the grasp's hand configuration and its local region of 3D points. A classifier will then be trained based on these features. In the actual grasping pipeline, where a set of candidate grasps will be given, the features of each grasp's local region will be computed, and the classifier will then predict which grasps are good and score each of the grasps. The candidate grasp with the highest score will then be executed by STAIR. Algorithm 1 summarizes the pipeline for STAIR grasping; the work of this project is mainly in the second for-loop.

Algorithm 1 Grasping an Object with STAIR

1: acquire 3D point cloud of environment using camera (SwissRanger)
2: get candidate grasp points using grasp point identification algorithm [1]
3: **for** each grasp point in candidate grasp points set **do**
4: use arm inverse kinematics to generate configuration(s) with hand at grasp point
5: use PRM path planner to generate path(s) to configuration(s) (if possible) [2]
6: add valid path(s) to candidate grasps set
7: **end for**
8: **for** each path in candidate grasps set **do**
9: extract local point cloud of the hand's end configuration
10: compute features using local point cloud and hand configuration (see Section 3)
11: use features to classify and predict if grasp will be good/bad
12: $Score[grasp] \Leftarrow ScoreForCandidateGrasp$ (from classifier)
13: **end for**
14: execute grasp = $\arg\max Score[grasp]$

3 Features

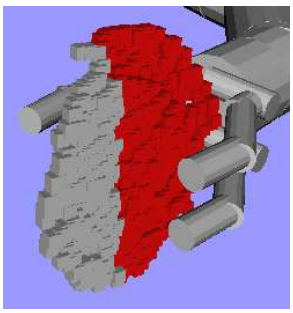
A total of 19 features, under 2 main categories, were most effective on the training data. A standard classifier was then trained using these features; a logistic regression classifier was the most accurate, and the value of the sigmoid function is used as the score.

3.1 Local point cloud distribution

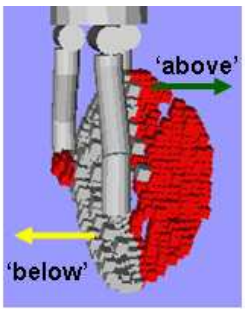
The placement of the object in the grasp is crucial to grasping success. If the object is nonexistent or is unevenly distributed in the grasp, the grasp may be unstable. The first is measured simply by counting the number of points in the local region point cloud. Intuitively, more points means a bigger object to grasp, which generally decreases the difficulty of grasping it (less likely to miss). Just counting this region however is insufficient, as an object may be near the hand but is not in the grasp (since the region is larger than the hand's grasp). Hence the points in the actual grasp region, i.e., on the inside region of the fingers, are also counted. The last region that was counted is a special 'edge' region, defined as all points in the region not extending further than the out-most fingertip. This region usually defines the edge of the object, hence the given name (see Figure 1(a) for example).

Even if there are many points near/within the hand, its distribution is also important. For example, it is preferable to grasp a stick at the middle rather than the tip, as slippage is easier in the latter case. When considering the local or edge regions as defined above, this corresponds to the points 'above' and 'below' the hand (see Figure 1(b) for example). For planar objects, an even distribution (1:1 ratio) on both sides is desirable, i.e., it is not grasping a tip/corner. Counting can be done on points that are above/below the center, or on points that are strictly above/below the hand. The latter is more harsh, but can be more pivotal. Hence these two features are computed for each of the local and edge regions:

1. Evenness about center: $|\frac{1}{2} - \frac{Points\ above\ of\ center}{Total\ \#\ of\ points\ in\ region}|$
2. Evenness strictly above/below hand: $|\frac{1}{2} - \frac{Points\ above\ of\ hand}{Points\ above\ of\ hand\ +\ Points\ below\ of\ hand}|$



(a) Edge region points



(b) Points strictly above/below hand

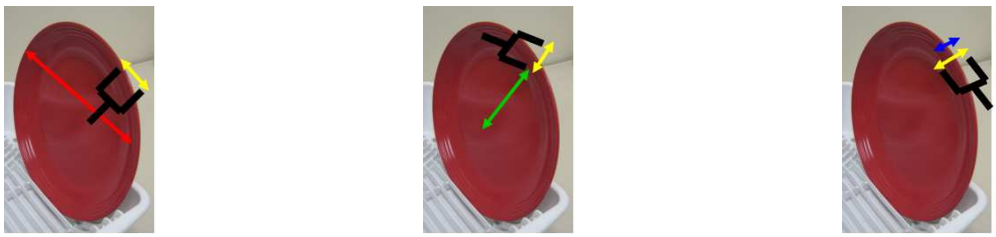
Figure 1: Features on local point cloud distribution (counted points in red)

3.2 Local plane approximation

The orientation of the of a grasp is just as important as its positioning; if the hand is placed at an incorrect angle, the grasp will be unstable and may miss the object or easily induce slippage when picking it up. It is desirable to grasp at narrow sides of an object, as the most force can be applied at these locations to achieve a tight closure on the object. Moreover, grasping on wide sides is extremely undesirable, as these sides may be wider than the hand.

To formalize this, let the *hand direction* be defined as the direction between the two sides of the hand (in a pincer grasp). Then it is desirable that in this direction, the object is 'narrow'. In other words, it is desirable that the (approximately locally planar) point cloud region has a plane normal that is parallel to the hand direction, since the plane normal is the direction in which the plane is least 'significant'. In general, the more 'significant' the direction of the plane, the less it is desirable that the hand direction be parallel to this direction. This can be approximated by taking the singular value decomposition of the 3D point cloud represented as a N*3 matrix, where N is the number of points. This gives three singular values with their corresponding orthonormal component directions; the larger the singular value, the more 'significant' the component is in the point cloud.

An example of these three components can be found in Figure 2 below; as demonstrated by the black 'hand', a grasp at the edge of the plate in the first (rather 'significant') two components will cause collision into the plate and thus failure; the only hand direction that is likely to give a good grasp is parallel to the (smallest) third component, the plane normal.



(a) 1st dir'n (b) 2nd dir'n (c) 3rd dir'n

Figure 2: Grasping in principle component directions (hand direction in yellow)

The metric used for measuring directional similarity was the absolute value of the dot product between the hand direction and each component direction; the absolute value was used as whether the directions are parallel or anti-parallel does not matter. A separate metric that accounted for the singular value of the component was also used; ideally, the directions should be orthogonal for large singular values, and parallel/anti-parallel for small singular values. Hence, the difference of the actual absolute dot product value from the ideal value (given the component's singular value) was taken. These two metrics are computed for the three component directions, using both the local and 'edge' regions, giving 12 features total:

1. Directional similarity: $|Unit\ component\ direction \cdot Unit\ hand\ direction|$
2. Difference from ideal: $(\frac{Largest\ singular\ value\ -\ Component\ singular\ value}{Largest\ singular\ value\ -\ Smallest\ singular\ value} - Directional\ similarity)^2$

4 Results

The training set comprised of 200 hand configurations and their respective local region point clouds acquired from the laser scanner camera. Half of these were hand-labeled as good, and the other half bad. These configurations corresponded to potential grasps on 8 different objects, 2 from each of 4 different object classes (plates, bowls, cylindrical cups, wooden blocks). The features described were computed for these data samples, and 10-fold cross-validation was used to evaluate the model learned using logistic regression. The average test set accuracy was 85%, and the average training set accuracy was 90%. This is on par with the grasp point identification component of the grasping pipeline.

The errors consisted of 14% negative errors (false positives) and 16% positive errors (false negatives), giving an average error of 15%. On a more positive note, the negative errors were mostly marginally positive, i.e., their scores were just above the margin, indicating low confidence levels. As most good grasps have high scores, as long as there exists a good grasp in the candidate set, then marginal negative errors will not be chosen by the pipeline.

The entire grasping pipeline with these features and the learned classifier was tested on STAIR 2, with a 75% accuracy on 20 attempts. This is an improvement from a roughly 50% accuracy on grasping if the grasp was randomly chosen from the candidate set of grasps.

5 Conclusion and Future Work

Using features from the hand's local point cloud of candidate grasps, the trained classifier performed reliably in predicting and selecting a good grasp in both the training set and the actual grasping pipeline. This result, however, is currently limited to several classes of large objects; more tests on other object classes are necessary in the future.

One limiting problem is that the grasping pipeline is reliant on inverse kinematics to provide candidate grasp configurations, which does not use information related to the quality to the grasp, and often limits the quality of candidates. Instead, an *efficient* search can be performed around identified potential grasp points, and using an objective related to the designed features, choose a grasp configuration that performs best on the objective.

6 Acknowledgments

This project would not have been possible without all members of the STAIR Perception-Manipulation team and their efforts to develop and expand the functionality of the STAIR robots. Special thanks also to Ashutosh Saxena for providing guidance for this project.

References

[1] A. Saxena, J. Driemeyer, and A. Ng, "Robotic grasping of novel objects using vision," *IJRR*, 2007.

[2] A. Saxena, L. Wong, M. Quigley, and A. Ng, "A vision-based system for grasping novel objects," *Unpublished manuscript*, 2007.