Learning to Select Robotic Grasps Using Vision on the Stanford Artificial Intelligence Robot

Lawson Wong[1]

Grasping is an essential ability for manipulation; for robots such as the Stanford Artificial Intelligence Robot (STAIR) to be resourceful in real-world environments, they must know how to grasp. While this is a well-studied problem in the case when a full 3-D model of the target object is known, it is difficult for real-world scenarios, where the robot must rely on imperfect perception to model the scenario. This paper presents a novel approach for grasping that only uses local 3-D information acquired from sensors. Given data of the environment from 3-D sensors, our algorithm generates arm/hand configurations that may potentially achieve a good grasp, then computes features of these candidates to select the best candidate and execute its grasp. These features capture desirable properties of potential grasps based on sensor data, which our learning algorithm then uses to predict how likely the grasp will be successful. This algorithm was tested on STAIR in real-world grasping of single objects and of objects in cluttered environments. Significant improvements in both cases were found.

Introduction

As the field of artificial intelligence becomes increasingly advanced and integrated, it is time to revisit the half-century old "AI-Dream," where intelligent robotic agents were envisioned to interact with the general human population. To this end, the Stanford Artificial Intelligence Robot (STAIR) project aims to introduce robots into home and office environments, where they will facilitate and cooperate with people directly. In order for robots to have any non-trivial use in such environments, they must have the ability to manipulate objects, which is provided through robotic arms. An arm usually has a manipulator "hand" attached at the end to allow finer manipulation and, more importantly, grasping. The ability to grasp is crucial; if we were unable to grasp with our hands, we would find it very difficult to perform essential tasks such as eating, and more complex actions such as cooking and working in an office would definitely be unachievable. A robust and infallible grasping system is therefore necessary for STAIR to achieve its goal.

In this paper, a novel approach for robotic grasping will be discussed. By considering information acquired from our 3-D visual sensors, we developed a reliable and efficient grasping system for STAIR that works in unknown and cluttered environments.

Background

The problem of robotic grasping has existed and has been well studied over the past few decades. The conventional approach use the forces applied by the fingers on the object at their contact points to determine whether a stable grasp can be achieved[1]. While in theory this fully determines the result of the grasp, this approach is not practical because a complete and precise model of the target object is necessary. If the model was inaccurate, force computations would likely be incorrect. When working in unknown and dynamic real-world environments, STAIR can only acquire a model of the environment through visual perception, which is subject to inaccuracies and incompleteness. In practice, applying force computations directly on these models leads to poor results.

The limitations imposed by perception have spurred interest over the past two decades in vision-based grasping systems. In particular, it has been found that perception of 2-D planar objects usually suffers from fewer problems. For such objects, the object

Figure 1: STAIR grasping from a very cluttered environment.

[1]Stanford University

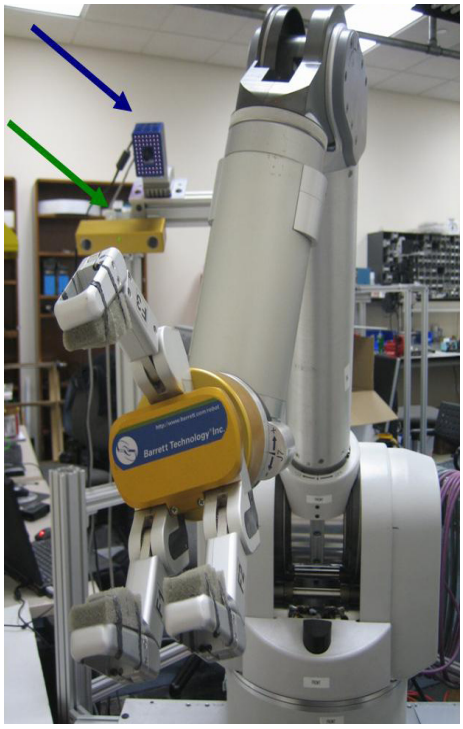

Figure 2: STAIR. 7-dof Barrett WAM Arm and 4-dof 3-fingered BarrettHand with "open" spread pictured. The spread can be "closed" such that all 3 fingers will be at the top. Vision system mounted on robot frame; blue arrow marks SwissRanger, green arrow marks Bumblebee2.

surface contour can be found reliably from vision. Criteria for successful grasps, derived from the mentioned theoretical force computations, can then be found for the object[2,3]. A similar approach was used by Kamon, Flash, and Edelman, where features indicative of successful grasps were computed given a 2-D image of the object[4]. A learnt model then used these features to compute an overall grasp quality, which predicted whether a grasp would succeed or not. While their results are promising, the methods are limited to 2-D objects and generalize poorly to the 3-D scenarios that STAIR faces.

Robot Description

The STAIR robot that this project is targeted for consists of a 7-dof arm (WAM, by Barrett Technologies) situation on a mobile platform. The arm is equipped with a 3-fingered hand with 4 degrees of freedom, one for each finger and one for the spread of the fingers (varying between being adjacent to each other and where two fingers are opposite the middle finger) (see Fig. 2). The arm is capable of reaching objects within a 1m radius. The hand can close its fingers inwards until the fingers hit an object, which is useful for grasping.

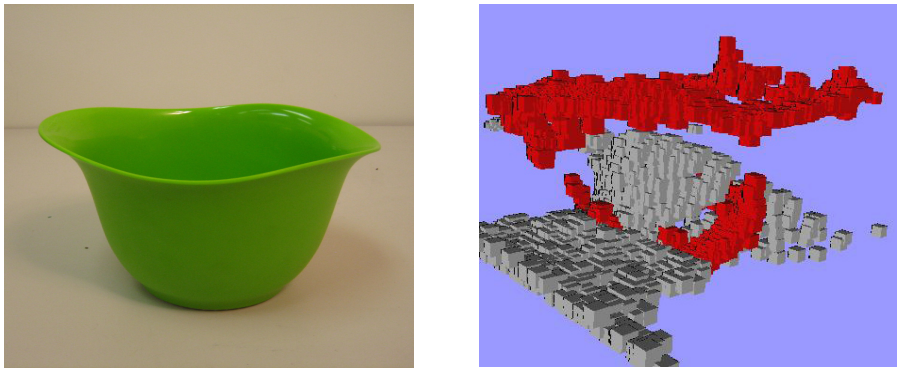

Figure 3: Imperfect perception. Original bowl, and the point cloud obtained via vision (shown in simulation). Red points come from Bumblebee2, gray points from SwissRanger. Only some edges are picked up by the Bumblebee2, and neither the bowl surface or table is seen. The SwissRanger gives a much more complete bowl front face and table, but no other side of the bowl is seen. Interestingly, the two cameras complement each other in this scenario; however, the perception of the bowl is still far from complete.

STAIR is also equipped with two cameras mounted on the robot frame. A stereo camera (Bumblebee2, by Point Grey Research) captures a 640*480 image using both its lenses, and uses the image differences to compute the depth for each image pixel, thereby giving 3-D point information. We shall refer to the set of 3-D points returned by the camera as the scene's "point-cloud." The point-cloud returned by the stereo camera is very incomplete, as stereo correspondences cannot be found for regions without texture such as object surfaces and tabletops, and only the front face of objects can be detected (the back face is occluded). To compensate for this missing information, another camera (SwissRanger, by MESA Imaging) provides a 144*176 array of depth estimates by firing an infrared light source and measuring the time it takes to reflect back to the camera. While this gives a much more complete image of the scenario, the data points are relatively sparse, and object surfaces that absorb or scatter the light are undetected by the camera. While the point clouds from STAIR's vision system are relatively accurate, they clearly still suffer from large amounts of missing data, hence an approach that does not apply force computations to evaluate grasps is necessary (see Fig. 3).

Approach

The objective is to, given a model of the environment through visual perception, determine a robot configuration (joint angles for the arm and hand) such that, when closing the fingers at that point (until they are fully closed or they hit an object), some object in the environment is successfully

Figure 4: 2-D image-based classifier identifying potential grasp points (depicted in red squares)[5,6].

grasped. This configuration shall be denoted as a "grasp." A successful grasp is defined here to be where the object can be lifted up into the air (such that the table is not supporting it) without it falling out of the hand.

We split this problem into two parts. We first find a set of likely candidate configurations that may achieve a good grasp, then use features of these candidates and a learnt model to score each of the candidates, and finally execute the highest scoring grasp. The first component has already been addressed by previous STAIR work on grasping[5,6,7]. Specifically, a 2-D image-based classifier uses the images from both cameras to select a set of corresponding 3-D points that are likely to be good grasp points (see Fig. 4). Given a 3-D point, there are still many orientations at which the arm can reach that point (and very few result in successful grasps), hence orientations are uniformly sampled for each point. These point-orientation pairs are then converted to joint angles, giving the corresponding robot grasp configuration. This forms our candidate configuration set.

The second component is inspired by the work of Kamon, Flash, and Edelman as described in the background section, where features of grasps are extracted and used to determine the "quality" of a grasp. The motivation behind this is that while force computations on perceived objects do not perform well and are inefficient, there are local properties of a grasp that inform us whether grasping at that location will be successful. There are several advantages to using local information. First, the most important 3-D region to consider for grasping is the region where the grasp will occur; little can be gained by considering the ends of a stick that we grasp in the middle. Second, while the vision data is incomplete, its *distribution* of incompleteness is very skewed; a bowl will have most of its front face perceived by the SwissRanger, but most of the back half would be missing. Hence we can get a more complete model when we grasp at the front face. Finally, there are usually much fewer points in the local region, which significantly speeds up computation. The previous work was limited to 2-D information, hence more sophisticated features, as described in the next section, will be computed using our 3-D local point cloud. A supervised learning algorithm will then be used to train a classifier based on these features, which can then be used to predict a score between 0 and 1 of the quality of a candidate grasp.

```
1.  Acquire 2-D candidate grasp points set from camera images using classifier[5,6,7]
2.  Use camera depth information to find corresponding 3-D candidate grasp points set
3.  FOR each grasp point in 3-D candidate grasp points set DO
4.      Sample orientations from 3-D orientation space
5.      FOR each orientation sampled DO
6.          Use arm inverse kinematics to generate configuration with hand center near the
            3-D grasp point and satisfying the 3-D orientation chosen.
7.          Select a finger configuration (sample spread and finger opening) that does not
            result in arm and hand colliding with obstacles
8.          Add arm/hand configuration from 7 (if any) to candidate configuration set
9.      END FOR
10. END FOR
11. FOR each configuration in candidate configuration set DO
12.     Compute features using the configuration and its hand's local point cloud
13.     Score[grasp] := score from classifier given features from 12
14. END FOR
15. WHILE grasp not executed AND candidate configuration set not empty DO
16.     grasp* := argmax Score[grasp]
17.     Plan path to execute grasp* using Probabilistic Roadmap motion planner[9]
18.     IF plan successful THEN execute plan
19.     ELSE remove grasp* from candidate configuration set
20. END WHILE
```

Table 1: Algorithm for grasping an object

The described procedure for grasping an object is summarized in the algorithm in Table 1.[8]

Features

Three main properties of grasping were considered. First, the grasp must be able to achieve good contact with the target object, otherwise the object may be entirely missed by the hand. Second, the grasp should be stable, so in particular an object should not be grasped at a tip or corner when that is unnecessary. Third, the grasp must be able to apply forces on the object effectively, which is dictated by the direction and orientation of the grasp; for example, consider grasping a long tube along its axis versus perpendicular to its axis. A total of 19 features were developed under these three categories.

The contact between the hand and the object can be approximated by presence of point-cloud points inside the hand. Intuitively, the more points within the volume of the hand, the bigger the grasping area and volume of the object, hence the less likely a miss will occur. Similarly, if there are very

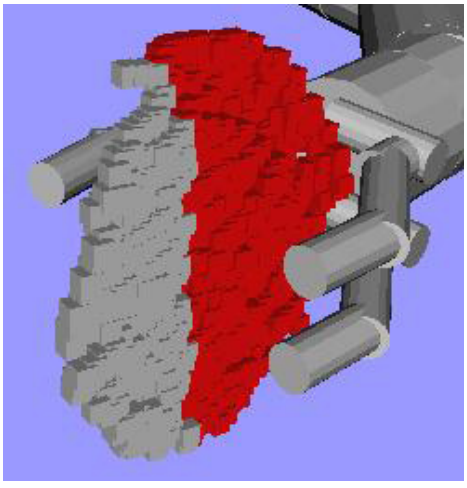

Figure 5: The cubes represent the local region. The red points within the local region denote the edge region.

few points within the hand, the grasp may likely fail because the points may just have been noise (where the hand will grasp air) or may have been a small tip of the object (where the hand should grasp some other part of the object). We therefore simply count the number of points within the local region, defined to be a sphere with 10cm radius centered at the hand center. 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 local region not extending further than the fingertip's reach (see Fig. 5). This region usually defines the edge of the object, hence the given name. Note that this feature has certain drawbacks, as small objects will naturally have fewer points but should not be undesirable to grasp; such subtleties are accounted for by the training set and the learning algorithm.

Stability of a grasp depends on the distribution of the object within the hand, or in our case, the distribution of the point cloud. Ideally, about the center of the hand, the point cloud should be evenly distributed along all axes. The outward axis from the hand is accounted for by the previous feature; if not enough points are within the hand, especially within the edge region, the grasp will be marked as bad. The "horizontal" axis, defined to be the axis between the fingers (when the outer fingers are directly opposite the middle finger), is not too important. A skewed distribution along this axis means that when closing the fingers to grasp, the closer finger(s) will push the object towards the farther finger(s), which is not a problem. The final "vertical" axis, which is normal to the other two axes, needs to be accounted for. Denoting one side of this axis about the center as "above" and the other "below," we desire that the number of points above and below the hand center to be near a 1:1 ratio (see Fig. 6). We therefore compute this feature by

$$\left| \frac{1}{2} - \frac{\text{Points above}}{\text{Points above} + \text{Points below}} \right|$$

, which is the absolute difference between the ideal (where points above = points below) and actual distributions. We also consider a similar measure where we only count points strictly above and below the hand (not enclosed by the hand). These measures are also computed with both the local and edge regions to increase robustness towards different cases; for example, the second measure may be more useful when considering large objects. The previous feature category combined with this therefore account for grasp stability.

Apart from being stable, it is more important that the forces of a grasp must be applied effectively on the object. Intuitively, an object should be grasped at narrow sides and not at wide sides, as at narrow places a tight closure on the object can be easily achieved, whereas at wide sides this is difficult (if the side is wider than the hand, then it is impossible). To capture this intuition, we consider the principal components of the local and edge regions. Using singular value decomposition (SVD), we obtain three orthonormal component directions u_i with variances σ_i, with σ_1 largest and σ_3 smallest. The larger the variance, the more important the direction is in defining the region; for example, for a plate, u_1 and u_2 will lie on the face (with large σ_1 and σ_2), whereas u_3 will be normal to the plate (small σ_3) (see Fig. 7). If we consider the unit horizontal axis vector h (axis running between the fingers), which is the direction in which the fingers close, we want h to be parallel to directions with small variances, and orthogonal to those with large variances. We therefore compute the directional similarity

$$s_i = |u_i \cdot h|$$

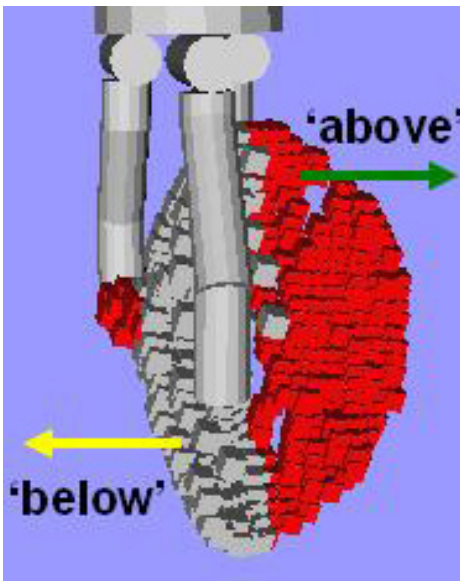

Figure 6: Definition of being above and below the hand. Red points denote regions strictly above and below the hand (not enclosed in hand).

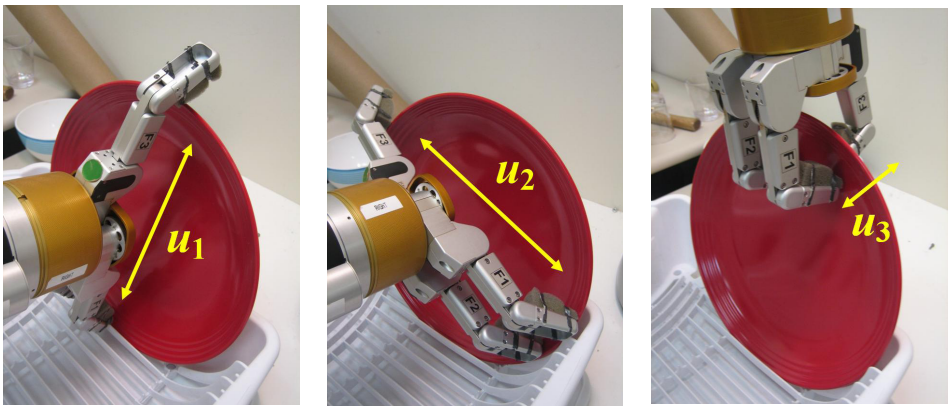

Figure 7: Example of principal component directions of plate. u_1 (left), u_2 (middle) lie on the plate, whereas u_3 (right) is normal to the plate. Only this direction gives good grasps.

for each component direction, which is large when u_i and h are parallel. Hence we desire that s_1 be 0 and s_3 be 1. We therefore measure this by computing the difference between the directional similarity and its ideal value by

$$\left(\frac{\sigma_1-\sigma_i}{\sigma_1-\sigma_3}-s_i\right)^2$$

Depending on how large σ_2 is, it may or may not be desirable to grasp in the direction of u_2. These features therefore capture whether the grasp configuration has a good orientation.

The features from all three categories were computed for a training set of 300 grasps, consisting of an equal number of good and bad grasps on plates, bowls, and wooden blocks, and achieved an 85% average test set accuracy when using 10-fold cross validation.

Experimental Results

We first considered grasping single objects from 13 novel classes (i.e., of different types from the training set) in a total of 150 experiments. These objects also differed significantly in size, shape, and appearance. In each trial, one object was placed randomly on a table in front of the robot. STAIR was able to achieve an overall grasp success rate of 76%, which is an improvement from the 70% achieved previously[9]. Moreover, the success rate was much higher at 86% for objects that were 1.5-3 times the size of the hand.

We also conducted grasping experiments in cluttered environments, which was the main objective of this project. In a total of 40 experiments, where more than 5 objects were placed randomly but close to each other, STAIR had to avoid hitting other objects and grasp a single object from the scenario. Although this was a significantly harder task in terms of perception, manipulation, and planning, STAIR had a success rate of 75%.

The videos and results of the experiments are at: http://stair.stanford.edu

Conclusion

We presented a robust and efficient algorithm that, given a 2-D image and 3-D point cloud of the environment from STAIR's vision system, can generate candidate grasp configurations and use local point cloud features to select a good grasp. The algorithm has been tested in simulation and in real-world experiments on STAIR, and has achieved significant improvement compared to previous systems, especially when grasping in cluttered environments. To further improve the algorithm, more features that describe general properties of grasps should be developed, and more grasp candidates should be searched and evaluated to increase the chances of finding and selecting an optimal grasp. In particular, instead of randomly sampling hand orientations uniformly from 3-D orientation space, better candidates can be found by applying heuristics to prune the search space. Eventually, we also hope to provide STAIR the sense of touch via force feedback, which would be extremely helpful in determining whether a secure grasp has been made yet. The challenge is to integrate all these components into a robust system without compromising for efficiency.

Acknowledgments

More details of the algorithm and results can be found in Saxena, Wong, Quigley et al.[6], Saxena, Driemeyer, and Ng[7], and Saxena, Wong, and Ng[8]. 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 and Professor Andrew Ng for providing guidance for this project.

References

1. Bicchi A, Kumar V. Robotic grasping and contact: a review. IEEE Intl Conf on Robotics and Automation Proceedings 2000; 1:348-353.
2. Morales A, Chinellato E, Sanz PJ et al. Learning to predict grasp reliability for a multifinger robot hand by using visual features. Intl Conf on AI and Soft Computing 2004.
3. Chinellato E, Morales A, Fisher R et al. Visual quality measures for characterizing planar robot grasps. IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews 2005; 35:30-41.
4. Kamon I, Flash T, Edelman S. Learning to grasp using visual information. IEEE Intl Conf on Robots and Automation Proceedings 1994; 3:2470-2476.
5. Saxena A, Driemeyer J, Kearns J et al. Robotic grasping of novel objects. Advances in Neural Info Processing Systems 2007; 19:1209-1216.
6. Saxena A, Wong L, Quigley M et al. A vision-based system for grasping novel objects in cluttered environments. Intl Symposium of Robotics Research Proceedings 2007.
7. Saxena A, Driemeyer J, Ng AY. Robotic grasping of novel objects using vision. Intl Journal of Robotics Research 2008; 27(2):157-173.
8. Saxena A, Wong L, Ng AY. Learning grasp strategies with partial shape information. Assoc for Advancement in AI Proceedings 2008.
9. Schwarzer F, Saha M, Latombe JC. Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. IEEE Trans on Robotics 2005; 21(3):338-353.

LAWSON WONG is a junior and coterminal master's student at Stanford University majoring in computer science (with honors), and specializes in artificial intelligence. He hopes to ultimately understand what intelligence is and how to algorithmically replicate it, and currently plans to pursue a PhD in machine learning. Before studying at Stanford, Lawson spent his entire life in Hong Kong, where he developed a passion for mathematics, physics, and logic that remains till today and occupies his time outside of computer science. He thinks that undergraduate teaching and research are extremely valuable and enriching learning experiences, and he thanks Professor Andrew Ng and Ashutosh Saxena for their guidance on the STAIR project. More information about Lawson can be found at http://www.stanford.edu/~lsw/.