ROBOTIC GRASPING ON THE
STANFORD ARTIFICIAL INTELLIGENCE ROBOT

AN HONORS THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
OF STANFORD UNIVERSITY

Lawson L.S. Wong
Principal Adviser: Andrew Y. Ng
May 20, 2008

Contents

Abstract

The STanford Artificial Intelligence Robot (STAIR) project aims to construct and develop robots that can work in real-world environments such as homes and offices, where they will interact regularly with people. To work in such places, manipulation is an essential ability, and one fundamental skill of manipulation is grasping. Robotic grasping has therefore long been a focus of the STAIR project. Because STAIR needs to operate in dynamic and unknown environments, no model of the world or objects to grasp is available; instead, it must perceive all this from its visual sensors. This makes grasping a very difficult problem because data from visual sensors is noisy and incomplete. A robust grasping system for STAIR using only visual input is therefore presented here.

One approach used in the past to solve the grasping problem is to separate it into two components, the first for finding feasible robotic arm/hand grasp configurations, and the second to select the best one of these configurations as the final grasp to execute. This is known as the comparative approach. We build upon and generalize this approach. For the first component, which we denote as "search," a previously developed image-based classifier that we developed is used to find plausible grasping points, then a search strategy is applied to find feasible grasp configurations. For the second component, which we denote as "selection," we notice that there are some properties of a candidate grasp configuration that are indicative of good/bad grasps, and that these properties remain consistent across different objects. Moreover, these properties can be easily and reliably computed even if the 3-D point cloud from visual sensors is incomplete. We therefore present a selection algorithm that computes these features for candidate grasp candidates, and using a trained logistic classifier produces a quality score for each grasp. The grasp with maximum score is then executed.

Extensive experiments using this grasping system were performed on STAIR. In particular, reasonable performance was found in all three tasks assessed: grasping single objects of various appearances, shapes, and sizes, grasping objects in cluttered environments, and applying the algorithm on a separate robotic platform to unload items from a dishwasher. Performance was comparable in each case, which indicates that the grasping system is effective in its targeted environments and is generalizable to other robots.

Chapter 1

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 if STAIR is to achieve its future goals.

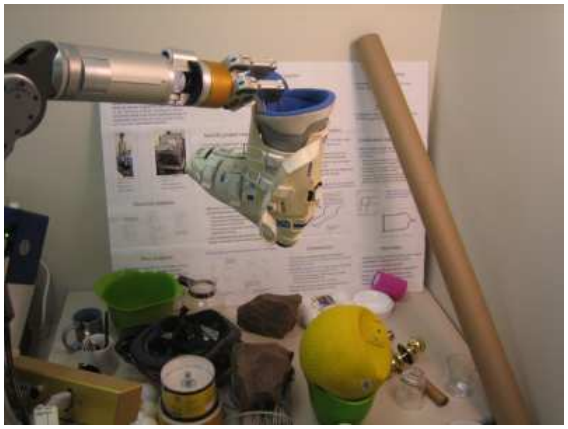


Figure 1.1: STAIR Grasping.

At first glance, the problem of grasping appears to be solvable using elementary physics. Given a 3-D model of an object, its surface properties, and the manipulator's kinematics, it is possible to derive conditions on the placement and forces of the manipulator's fingers such that the object will be stably grasped upon closing the fingers inwards. However, in most real-world environments the key piece of information is missing – an accurate 3-D model. Robots such as STAIR that operate in dynamic environments cannot rely on a static *a priori* model; instead, visual sensors must be used to perceive the world. Data acquired from these cameras are typically inaccurate and incomplete, which presents a significant challenge for the mechanical approach discussed as much of the object is actually missing from the data. As characteristic of many AI problems, they become much harder once the environment is not fully observable. In the context of real-world robotic grasping such as STAIR, the problem of grasping is therefore difficult and remains to be solved.

The objective of this work was to create a robust grasping system for STAIR that can operate in unknown and cluttered environments. The latter characteristic presents an extra challenge as the space of possible grasps becomes significantly limited by the presence of nearby objects; however, as it pertains to our target environments, we must consider it too. For this grasping system, full arm/manipulator kinematics have been assumed, and visual 2-D and 3-D data of the scene in front of the robot will be supplied by STAIR's cameras. From this data, the system is to provide a single goal grasp configuration $\theta \in \mathbb{R}^n$, where n is the number of joints that fully specifies the configuration of the arm/manipulator. (This goal configuration will sometimes informally be referred to as a "grasp.") This θ denotes a configuration at which, when the fingers are closed, should achieve a grasp. The system is to also provide and execute a trajectory that reaches θ and grasps the target object.

In this thesis, I will describe our grasping system, which is based on the comparative approach covered in Chapter 2. An overview of STAIR and our implementation of the approach will then be discussed in Chapter 3. The comparative approach consists of two steps, grasp configuration generation and selection, and these will be respectively covered in Chapters 4 and 5. Using this grasping system, we have performed many real-world grasping experiments, which are described in Chapter 6. Finally, conclusions drawn from these results and future directions will be discussed in Chapter 7.

Portions of this thesis have appeared in [1] and [2].

Chapter 2

Literature Review

A general overview of grasping approaches that use vision will be provided here. Note that many of the approaches discussed will not be applicable to our approach, with their limitations discussed in greater detail below. We will, however, build upon and generalize the work done in the comparative approach.

The analytic approach discussed above finds optimal grasping points and configurations, relative to some criteria for optimality based on the wrench matrix of a grasp, which is dependent on the normal force, tangential force, and moment about the normal at finger contact points [3]. As noted, such forces and moments are difficult to compute unless a full model of the object is present; simply applying these criteria to incomplete models leads to poor results in practice. Many variants of optimality criteria have been suggested, but they all assume full knowledge of the object's 3-D model.

Due to the limitations imposed by perception, much work has started to focus on grasping using visual information instead of using known models and environments. There are several major types of approaches in this area. One conceptually simple approach, proposed early on by Dunn and Segen, is to compile an object library with template grasps for each object [4]. The target object is then matched to one of the objects in the library via object recognition [5], and the stored grasp is then executed. While these methods are simple and viable, they are fundamentally limited by the knowledge base – an object that has not been seen before will often not lead to good grasps. Moreover, object recognition is yet another vast and active open area of research, so this approach to grasping has another difficulty to overcome.

Instead of identifying the grasp configuration itself, another approach is to find an optimal control policy for grasping the object. This allows the process to be adaptive, thereby taking into account the uncertainties in the environment. Anglani *et al.* used Q-Learning to train a controller that learned from errors and was able to grasp spherical objects identified through vision [6]. More recently, Hsiao, Kaelbling, and Lozana-Pérez used partially observable

Markov decision processes (POMDP) to capture the uncertainties in environment dynamics and sensor data [7]. Their system was able to find policies in simulation for two-finger grasping of simple objects. Although initial results appear promising, much more work is needed to extend this approach to complicated objects and to real-world environments.

An entirely different approach is the comparative approach, as termed by Kamon, Flash, and Edelman in [8]. This method splits the grasping process into two parts: grasp generation and evaluation. Candidate grasp configurations are first generated, then the quality of each grasp is evaluated, and the one with maximum quality is executed. The quality measure is computed based on a mixture of features, each of which is indicative of good/bad grasps and is easily computable from the object model provided by visual sensors. An example of such a feature is the angle between a finger's applied force direction and the normal of the contact surface, with the angle ideally being small (which reduces tangential slippage forces).

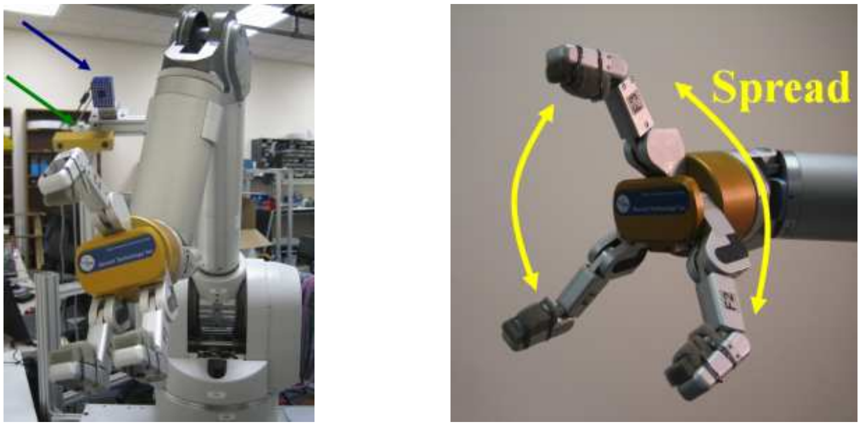
This approach was used by Pelossof *et al.*, who generated many grasp configurations for objects in simulation, then used Support Vector Machines (SVM) to learn and estimate the quality of each grasp from several parameters based on the shape of the object and hand orientation [9]. While this worked very well in simulation, it was not extended to real-world cases, which would be significantly more difficult because the shape and surface of the object would be unknown. Several works have tested their own systems outside of simulation, with a limited degree of success and generality. Kamon, Flash, and Edelman used estimations on the object's center of mass and symmetry axis to both generate and rank grasp configurations [8]. Despite their relatively simple list of features, their system was able to grasp most – even novel – objects; however, this was naturally limited to generally regular and symmetrical objects for which the center of mass can be rather accurately determined. Chinellato *et al.* and Morales *et al.* also jointly developed a grasping system that worked well for complex planar objects, using features based on the 2-D object contour [10, 11]. This was possible because for planar objects in uncluttered environments, the 2-D contour can be easily determined using vision, hence their object models were rather accurate; however, this does not generalize to our 3-D case. Although previous works using the comparative approach have yielded considerable successes in real-world grasping, the objects considered so far have been limited (regular, symmetrical, planar), and the methods used on these objects are not easily generalizable to complex 3-D objects. Moreover, little work has focused on grasping in cluttered environments, which make both grasp generation and feature computation more difficult. We will therefore develop more general methods using this approach.

Chapter 3

Approach

3.1 Description of Robot

In order to better understand our approach, I will first provide a description of STAIR. The STAIR robot that we use for grasping-related consists of a 7 degrees of freedom (dof) arm (WAM, by Barrett Technologies) situated on a mobile platform. The arm is equipped with a 3-fingered, 4-dof hand, one dof for each finger and one for the spread of the fingers. The spread allows the outer fingers to rotate around the hand synchronously, varying from being adjacent to each other (closed spread), to being opposite the middle finger (open spread) (see Figure 3.1b). The arm is capable of reaching objects within a 1m radius. Since there are $7 + 4 = 11$ degrees of freedom in the arm, grasp configurations θ are in $\mathbb{R}^{11}$.



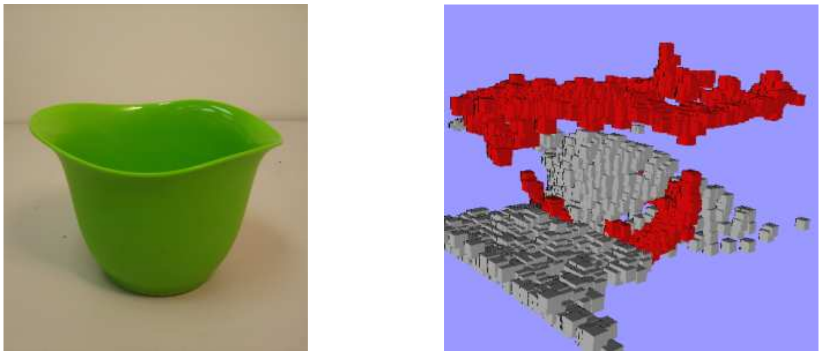
(a) STAIR Platform; green arrow points to Bumblebee2 camera, blue to SwissRanger

(b) BarrettHand on STAIR

Figure 3.1

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 smooth object surfaces and tabletops. Because objects usually have some untextured parts, the object's point cloud often has significant "holes" where stereo matches were not found. To compensate for this missing information, we included another camera (SwissRanger, by MESA Imaging) that provides a 144 × 176 array of depth estimates by time-of-flight measurements. Since this mechanism does not require matching images, the point cloud of an object's front face is generally complete (the back face is inevitably occluded). However, the data points are sparse, and object surfaces that absorb or scatter the camera light source cannot be detected.

While the point clouds from STAIR's vision system are relatively accurate, they clearly suffer from large amounts of missing data. Take, for instance, the point clouds of the bowl shown in Figure 3.2b. The Bumblebee2 point cloud (in red) picks up only the edges of the bowl, and neither the bowl surface nor the table is seen. The SwissRanger point cloud (in gray), on the other hand, gives a much more complete view of the bowl front face and table, but no other side of the bowl is seen. Although bowls are particularly difficult for both cameras to perceive, point clouds for most objects have similar properties and thus suffer from the same problems. Because of large amounts of missing data, none of the previously discussed approaches (and especially not the analytical one) work well for STAIR on general 3-D object grasping in cluttered scenarios.



(a) Actual bowl

(b) Point clouds of bowl

Figure 3.2: A bowl and its point clouds. Point cloud points are visualized as cubes; red cubes come from Bumblebee2, gray from SwissRanger.

3.2 Grasping System

As noted earlier, we generalize previous work using the comparative approach. To do so, we require a system that searches for grasp configurations (given the data from the two STAIR cameras) and a system that selects the best grasp configuration by extracting features from the 3-D object point cloud . The entire grasping system is shown in Figure 3.3.

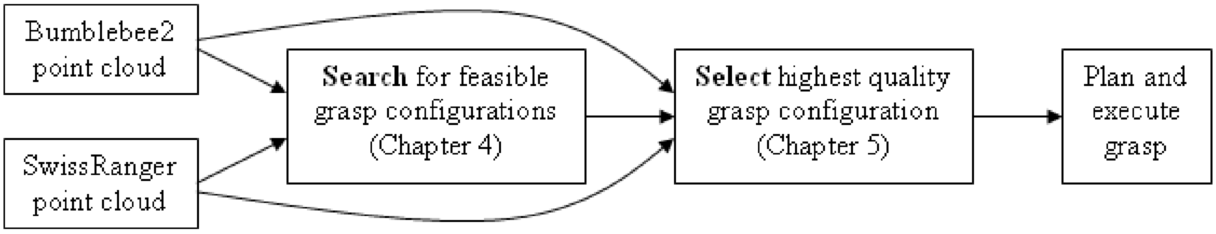

Bumblebee2 point cloud
SwissRanger point cloud
Search for feasible grasp configurations (Chapter 4)
Select highest quality grasp configuration (Chapter 5)
Plan and execute grasp

Figure 3.3: Grasping System

The search procedure has already been partially addressed by previous STAIR work on grasping [12, 13, 14]. Using synthetic images of various kitchen and office objects, features for small several-pixel wide square patches were computed to determine how likely that patch corresponded to a grasping point for the object. This was done for each patch in the image, giving a large feature vector for a single image. Several patches corresponding to plausible grasping points were then hand-labeled to be good points, and the rest bad. A logistic classifier was then trained using these features and labeled patches. When actual camera images are taken, features can then be computed for each patch, and the logistic classifier can then distinguish between good and bad grasping points based on its learned model. Each patch also receives a score, so the highest scoring patches are outputted as the most plausible grasping points in the image. From these 2-D image patches, the corresponding 3-D points in the point cloud can be found, which serve as the candidate targets for grasping. A similar classifier was trained for the SwissRanger camera, but directly uses the 3-D point cloud. We shall refer to this classifier as our image-based classifier, since it takes as input images from the camera and outputs plausible 3-D grasping points. Sample outputs from these classifiers can be seen in Figure 3.4. The grasping points from both classifiers are then grouped together and ranked, and the highest 20 3-D grasping points are used as targets. This has been the work so far. However, there is a large gap between these target grasping points in $\mathbb{R}^3$ and grasp configurations $\theta \in \mathbb{R}^{11}$. Extending target 3-D points into 11-D grasp configurations cannot simply be done by assigning arbitrary values; instead a search over the extra dimensions is necessary. This strategy will be discussed in Chapter 4.

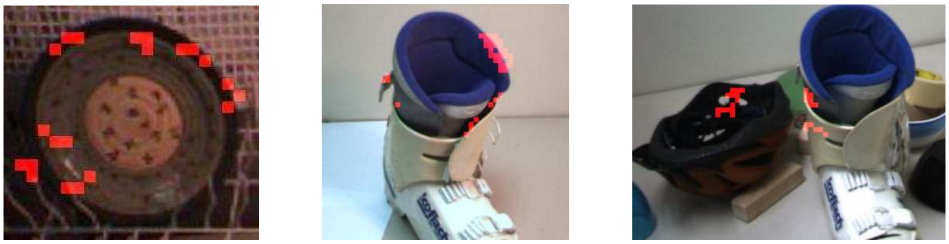


Figure 3.4: Sample outputs from image-based classifiers. Red patches indicate the predicted grasping points.

In contrast to the work already done on search, there has been little previous STAIR work on the selection procedure. Candidate grasp configurations were selected using simple heuristics, such as preferring higher grasp points (objects below are more likely to blocked by objects above) and selecting configurations where the last arm link is as close to a downward 45° angle as possible (which empirically worked well). In the past these heuristics often led to failure. Instead of using heuristics, we turn to the way in which previous works using the comparative approach have performed selection – features. Having features that capture properties of good/bad grasps allows us to identify, for a given grasp configuration and its features, whether the configuration will lead to a good grasp or not, and hence whether the configuration should be selected. We used supervised learning to train a logistic classifier for these features, such that given a configuration and its features, the classifier outputs a score for the quality of the configuration, the higher the better. The highest scoring configuration is then naturally selected. This approach will be discussed in Chapter 5.

Once a final grasp configuration is selected, what remains is execution. Given the start and end configurations in $\mathbb{R}^{11}$, we want to find a path in 11-D space of arm/hand joint angles that connects the two configurations without colliding into any obstacles modeled from the point-cloud points (these points, in this case, are also represented in 11-D space). The process of finding such a plan is referred to as motion planning. We use an efficient Probabilistic Roadmap (PRM) motion planner for this task [15]. In PRM motion planning, points in the 11-D joint angle space are randomly sampled, then all obstacle free segments between points are found and form the "roadmap" of the space. Any graph search algorithm can then be applied to find a path from the source to the goal. If such a path cannot be found, the next best configuration is chosen as the goal, and so on until a plan is found. When a plan is available, it is executed, and upon successful completion of the goal, the fingers are closed synchronously until they hit the object. A grasp is defined as successful if the object can then be lifted up into the air (with no support underneath) without its falling out of the hand. For more details on motion planning and grasp execution, see [13, 16].

Chapter 4

Search

We use as a baseline the 3-D grasping points returned from the 2-D and 3-D classifiers mentioned in Section 3. That is, for a given grasp configuration, we require that the end-effector location (i.e., the center point of the hand's palm) be at one of the 3-D grasping points identified. This requirement limits the dimensionality of the problem by 3 dimensions; hence, there are 8 dimensions left to search (4 for both the arm and the hand).

4.1 A Naïve Method

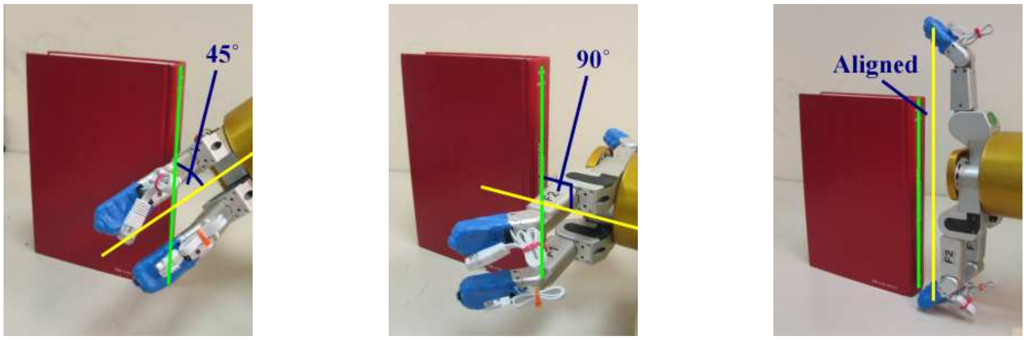
Given one 3-D grasping point, the search can be split into several subproblems. The first is the requirement that the end-effector must be at the 3-D point. From a 3-D point in Cartesian-space, we use the arm's inverse kinematics to find a 6-D joint-space configuration that reaches the 3-D point. Although the arm itself has 7 joints, because the final wrist roll joint only determines the rotational orientation of the hand, it does not affect the position of the palm. Hence the joint angles of the first 6 joints is sufficient to fully specify the end-effector position of the arm, and therefore inverse kinematics only needs to find values for only these 6 joints to satisfy the 3-D grasping point requirement. Because of redundancy, there are an infinite number of inverse kinematics solutions; in our naïve approach, the first solution found is used. The wrist roll and hand dimensions remain to be solved.

The second subproblem is the wrist roll. The simplest approach is to set the joint value as be the midpoint between its maximum and minimum limits; this allows the greatest flexibility in manipulating the object afterward. The third subproblem is the hand's spread; the simplest solution is to have a parallel-plate-like configuration, where the fingers close in onto each other. This is the case when the spread is fully open. The final subproblem is the finger openings; it intuitively makes sense to secure the maximal grasp possible, i.e., set the fingers as wide open as possible.

Combining the 6-D inverse kinematics solution and the fixed values for the wrist roll and all hand dimensions, we obtain a single naïve 11-D grasp candidate for the grasping point. Although this gives a reasonable and generally feasible solution, it is often a poor grasp candidate, with a suboptimal orientation and a poor hand configuration. We therefore use better search techniques to improve on these two issues.

4.2 Wrist Orientation

The orientation of the final arm link, or informally the "wrist," is crucial. This is because the orientation affects both the stability of the grasp and the directions in which forces can be applied by the fingers. For example, for grasping an upright book there is greater stability if the approach direction of the wrist is perpendicular to the spine (Figure 4.1a) than if it is at a downward 45° angle (Figure 4.1b), because the hand is then unable to tightly close around the book. Even small deviations from the optimal orientation can quickly lead to poor grasps. Clearly then, simply allowing inverse kinematics to arbitrarily pick an orientation for the arm as done in the naïve approach will likely lead to suboptimal configurations.



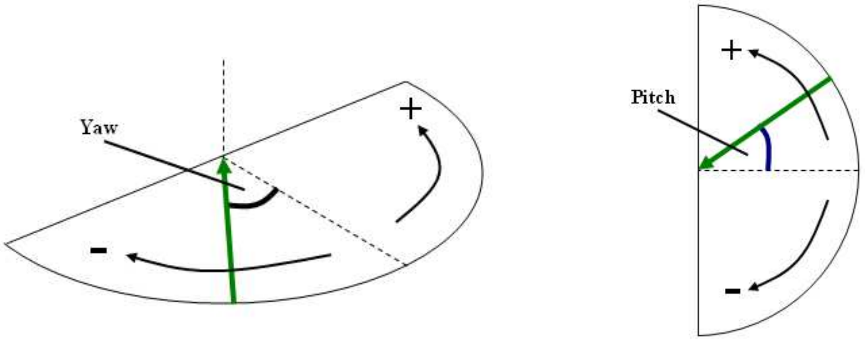
(a) Approach perpendicular to spine

(b) Approach at downward 45° angle

(c) Approach aligned with spine

Figure 4.1: Examples of good and bad grasps

We can specify the direction of the wrist using two parameters for orientation – the yaw and the pitch. A schematic for these is shown in Figure 4.2. The yaw determines the horizontal approach direction, and is 0 on the vertical plane; the pitch determines the vertical approach direction, and is 0 on the horizontal plane. Again, we do not consider the final orientation parameter – the wrist roll – as it does not affect the wrist direction. Using a combination of these two parameters, we can specify any wrist direction. Since the target grasping point only restricts 3 of the 6 dimensions searched by inverse kinematics, further restricting the dimensionality by 2 (by specifying yaw and pitch) is possible and in general



(a) Yaw

(b) Pitch

Figure 4.2

leads to feasible solutions.

To generate a representative set of configurations with different wrist orientations, we sample from the orientation parameters (yaw and pitch), then for each pair of parameters, the modified inverse kinematics returns several configurations that satisfy both position and orientation requirements. The selection of the set of orientation parameter pairs is simply a search over 2-D space. The full space for both yaw and pitch is $(-180°, 180°]$, and by discretizing the space, we can include all pairs into our search for good configurations. However, only $\frac{1}{8}$ of the space needs to be considered, since very few objects require grasping from underneath or behind. Moreover, since the cameras will not be able to see these regions as clearly as top and frontal areas, grasping from underneath or behind is considerably more dangerous. Hence, we only search over yaw in $[-90°, 90°]$ (in front) and pitch in $[0, 90°]$ (above). We discretize and sample parameter values with 30° steps.

4.3 Hand Configuration

The search for hand configurations includes several components. The most important one is the wrist roll, which has been excluded from the inverse kinematics search. As the middle finger cannot be moved around the hand, the wrist roll solely determines the finger's placement, which greatly affects grasping. For example, if the middle finger were aligned along the major axis of a book's spine, (Figure 4.1c), any grasp would have to close in along the spine's direction, which is difficult compared to grasping perpendicularly to the spine. As with the wrist directions above, we therefore generate configurations in steps of 30°, between the joint limits of $[-165°, 165°]$. Note that while the wrist link may appear symmetric, subtle differences in the fingers' placement prevent us from reducing the search space.

Search is less applicable for the actual hand parameters. The spread mainly determines the overall force directions. In general, the more useful type of grasp is the power grasp, where the spread is fully open. This applies strong forces in the directions that the fingers close in on. However, this grasp can be less stable because the force is concentrated in one direction. Hence a balanced spread, where the angle between any two adjacent fingers is the same, can sometimes be desirable, especially when a symmetric object such as a ball is being grasped. Only these 2 spreads are considered for each 7-D grasp configuration obtained from inverse kinematics and wrist roll search. Other spreads are either too similar, or are likely to fail (e.g., a fully closed spread would push away the object before a grasp can be made).

Finally, given the 8-D configuration from above, the finger openings are found. Although the intuition behind the naïve method, full opening of the fingers, is reasonable, it is likely to fail in cluttered environments because the fingers may hit adjacent objects. Instead of using the maximal opening, we therefore seek the minimal opening that gives a collision-free configuration. This ensures a feasible configuration (since it does not intersect any object according to the point cloud), yet provides a tight grasp and maximum safety from neighboring objects. For purposes of efficiency, we chose for all fingers to open to the same degree; allowing more freedom does not lead to large gains, and instead decreases the stability of grasps. Choosing the finger values therefore simply involves decreasing the opening from 1 (closed) to 0 (open) in steps of 0.1 until a collision-free configuration is found, at which point the configuration $\theta \in \mathbb{R}^{11}$ is added to the grasp configuration candidate set. If at a finger opening of even 0 the configuration remains in collision, then the configuration is simply infeasible and is discarded. Note that because collisions are checked using the motion planner's collision checker, we have the property that any grasp configuration in the candidate set is feasible, hence most impossible goal configurations are already pruned.

Using the components discussed, we can expand the search system shown in Figure 3.3 into the pipeline shown below in Figure 4.3.

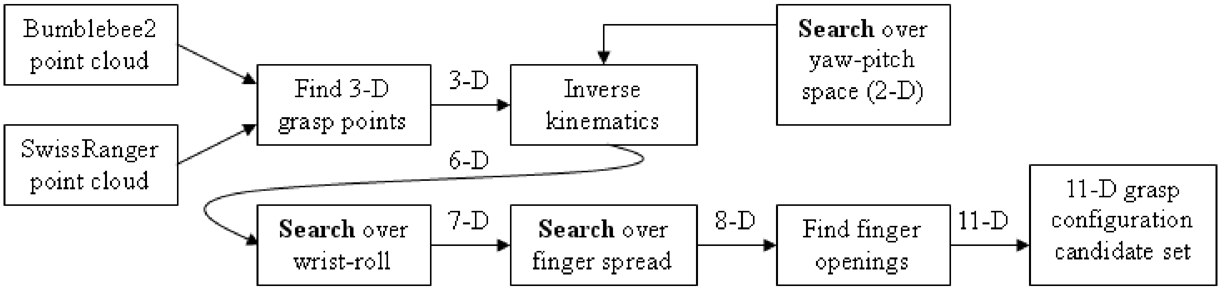


Figure 4.3: Search System

Chapter 5

Selection

In order to apply the selection part of the comparative approach, we must have a system that can rate a given grasp configuration's quality. This has been done in the past, as discussed in Chapter 2, by computing features that are indicative of good/bad grasps. In our case, we use the local 3-D point cloud of a grasp configuration to compute these features; the features will be discussed in Section 5.3. A set of actual point clouds were collected, and sample grasp configurations for these scenes were found. The configurations were also assigned a label 0 or 1, 0 indicating that it is a bad grasp, and 1 for a good grasp. The point clouds, configurations, and labels were then used as training samples to learn a model for a logistic classifier. The details of the training process will be described in Section 5.4. This model can then be used to evaluate the quality of future grasp configurations, as shown in Figure 5.1. From the candidate set of 11-D grasp configurations outputted by the search algorithm, the features of each candidate are computed, then a quality measure between 0 and 1 is assigned to each. The probabilistic interpretation of logistic regression suggests that this quality measure is equal to the probability that the configuration's label is 1, i.e., the probability that the configuration is a good grasp. Hence the configuration with maximum quality, i.e., the one most likely to be a good grasp, is selected.

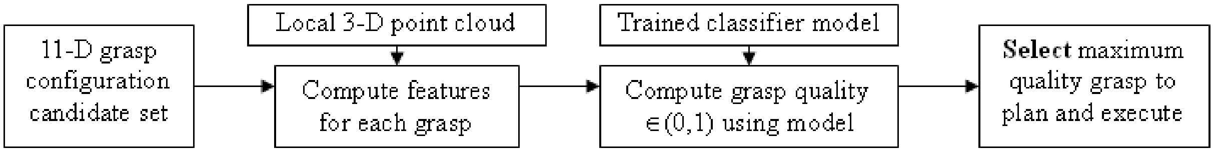

11-D grasp configuration candidate set
Local 3-D point cloud
Compute features for each grasp
Trained classifier model
Compute grasp quality ∈(0,1) using model
Select maximum quality grasp to plan and execute

Figure 5.1: Selection System

5.1 Motivation

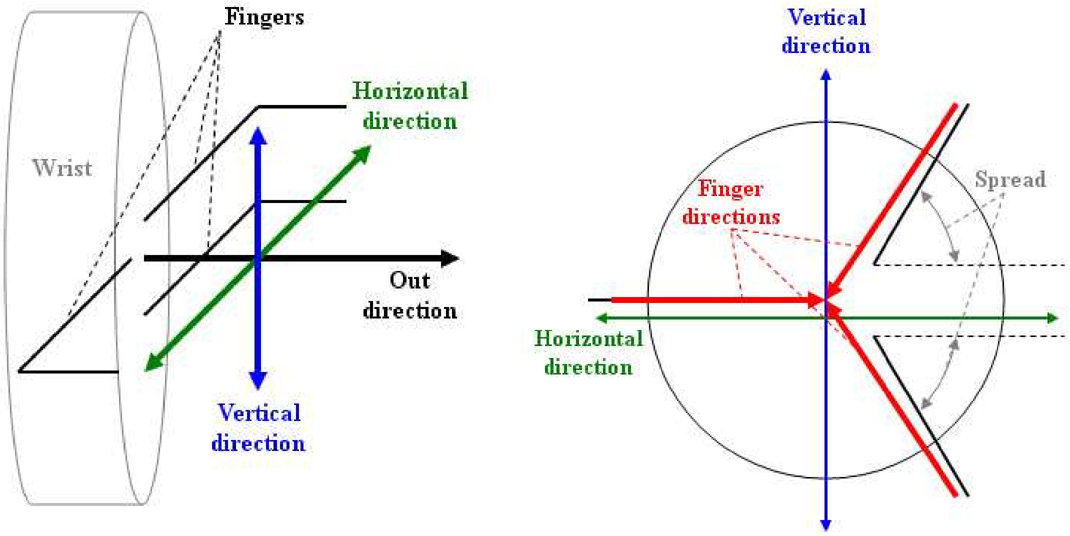
At this point it may seem strange that the comparative approach can actually work on the significantly incomplete sensor data. Previous works such as [8, 10, 11] worked around this problem by selecting scenarios where global object information could be reliably retained despite sensor errors. Examples of such information included a good estimate of the object's center of mass and symmetry axes, or even the object contour for planar objects. For general 3-D point clouds that STAIR normally receives, though, none of the above information can be accurately estimated. However, we observe that there are several point cloud properties that indicate that we do not necessarily need global object information. In particular, we hypothesize that it is sufficient in most cases to consider only the local region of an identified grasp location; if certain local properties are satisfied, then the grasp is likely to be successful.

Before the local properties are discussed, which essentially form our features, it is useful to understand the rationale behind this hypothesis. When given a 3-D location to grasp, the most important 3-D region to consider is the region where the grasp will occur, i.e., the local region. For example, little can be gained by considering the ends of a stick that we want to grasp in the middle. In addition, although the vision data is incomplete, its distribution of incompleteness is very skewed; an object will have most of its front face perceived by the SwissRanger camera, but most of the back half will be missing. Since the algorithm should always choose visible 3-D locations to grasp, we are therefore usually guaranteed a reasonably complete model of the local region. We can therefore achieve more reliable feature computations by considering only the local region, and without needing to discard too much useful information from external regions. As long as our features, to be described in Section 5.3, are local and try to capture good local properties in grasps, we claim that we can still accurately predict whether a given grasp configuration will be successful or not.

5.2 Definitions

Several definitions relating to hand configurations will become convenient when we describe our features, because they are mainly based on the hand configuration of the grasp. Note that the terms used here are not widely-used nomenclature, but are introduced solely for the purposes of clarity. The terms we will use most are (see Figure 5.2 for a schematic):

- Out vector/direction: This is the vector/direction that comes out from the center of the final arm link. The distance along this direction will be denoted as the "extension"; zero extension on the out vector denotes that the point is on the palm of the hand. The maximum extension of the fingertips is about 10cm.



(a) Side view

(b) Palm view

Figure 5.2

- Horizontal direction: This is the direction between the two sides of the hand, which is equivalent to the direction between the middle finger's tip and the point on the out vector with the same extension as the fingertip.

- Vertical direction: This is the direction normal to the out and horizontal directions. Distance along this direction will be denoted as the "height"; zero height is the height at the intersection of the out and horizontal directions. Being on either size of zero will be referred to as being "above" or "below" the hand; which side corresponds to which name is arbitrary (what matters is that there are two distinct sides).

- Finger direction: This is the direction in which the fingers close (and hence apply forces in). It is taken to be the vector pointing from the fingertip toward the point on the out vector with the same extension as the fingertip.

- Grasp center: This is the point on the out vector with extension 5cm. A sphere of radius 5cm about the grasp center, which we will call the grasp sphere, approximately defines the region that the hand can grasp.

- Region center: This is the point on the out vector with extension 10cm. A sphere of radius 10cm about the region center, which we will call the region sphere, is taken to be the local region; points further away usually do not yield significant information.

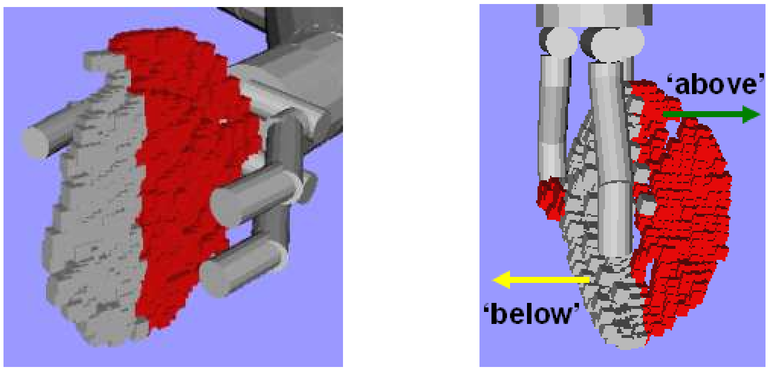
5.3 Features

Note that all features discussed here and previously by us in [1, 2, 17] can be easily computed from the local point cloud and are all generalizable to most robotic manipulators.

5.3.1 Presence

A grasp must first be able to achieve good contact with the target object; otherwise, the object may be entirely missed by the hand. This contact between the hand and the object can be approximated by the presence of point cloud points within 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 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 somewhere else). We therefore simply count the number of points within the grasp sphere.

Knowing that there are points nearby but not necessarily in the grasp sphere is useful too, as a large nearby presence increases the likelihood that an object is actually present. Hence the number of points within the region sphere and an "edge" region are also counted. The edge region is defined as all the points in the local region with extension less than the maximum finger extension; this usually defines the edge of the object, hence the given name. An example of the presence feature on the edge region is shown in Figure 5.3a. These counts give information about how the object appears near the target location and increase the robustness of the feature. Note, however, 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.



(a) Presence

(b) Stability

Figure 5.3: Presence and stability features. Red cubes in the point cloud are counted.

5.3.2 Stability

Assuming that an object is present, other grasp properties can be analyzed. Another important safety feature for grasp configurations is that it must be stable, so an object will not slip and fall out of the hand (hence by our definition of a grasp, the grasp fails). While general stability is difficult to determine without knowledge of the object's center of mass and other mechanical properties, local stability is a necessary condition for general stability.

Local stability of a grasp depends on the distribution of the object (i.e., the point cloud) within the hand. 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 presence feature; if not enough points are near the hand, especially within the edge region, the grasp will be marked as bad. The horizontal axis is not important because a skewed distribution along this axis simply means that when the fingers are closed for grasping, the closer finger(s) will push the object toward the farther finger(s), which in general is not a problem.

The final vertical axis needs to be accounted for. An even distribution along this axis means that the number of points above and below the hand should be equal (i.e., near a 1:1 ratio). We therefore compute this feature as $\left|\frac{1}{2} - \frac{\textit{Points above}}{\textit{Points above} + \textit{Points below}}\right|$, which is the absolute difference between the ideal (where points above = points below) and actual distributions. This formulation is used so that the level of idealness is monotonic, with 0 being the best, 0.5 the worst, and the lower the value the better the stability. We also consider a similar measure where we count points only strictly above and below the hand (i.e., not enclosed by the hand), again working from the notion that accounting for local regions neighboring the grasp sphere provides useful information. Figure 5.3b illustrates this strict feature on an example of an unstable grasp. These measures are computed with the region sphere and edge region, since stability depends more on the wider region than on the internal grasp sphere. The presence feature combined with the stability feature therefore accounts for grasp stability by considering the point cloud distribution in all three axis directions.

5.3.3 Grasp Direction

For an effective grasp, the hand must be able to apply forces on the object in the correct direction and orientation. For example, grasping a long tube along its axis is much less effective than grasping perpendicular to its axis. Intuitively, an object should be grasped at narrow places and not at wide sides because at narrow places a tight closure on the object can be easily achieved, whereas at wide sides this is difficult, or even impossible, if the side is wider than the hand. To capture this intuition, we consider the principal components of the region sphere, which denote the local shape and direction of the object.

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 greater the variance, the more important the direction is in defining the region; for example, for a plate as displayed in Figure 5.4, 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). If we consider the horizontal direction vector h, which is the direction in which the middle finger (and hence the hand) closes, we want h to be parallel to directions with small variances and orthogonal to those with large variances. In the case of the plate of Figure 5.4, if h were parallel to a direction with large variance (any combination of u_1 and u_2), the plate would be wider than the hand and no grasp would be possible. However, were h parallel to the plate normal (u_3), then a good grasp could be achieved.

We therefore compute the directional similarity $s_i = |u_i \cdot h|$ for each component direction. Since u_1 is largest and u_3 is smallest, it is desirable that $s_1 = 0$ and $s_3 = 1$. This is therefore measured by computing the difference between the directional similarity and its ideal value, which is given by $\left(\frac{\sigma_1 - \sigma_i}{\sigma_1 - \sigma_3} - s_i\right)^2$. Depending on how large σ_2 is, the above measure states that it may or may not be desirable to grasp in the direction of u_2. These features therefore capture whether the grasp configuration can effectively apply forces on the object.

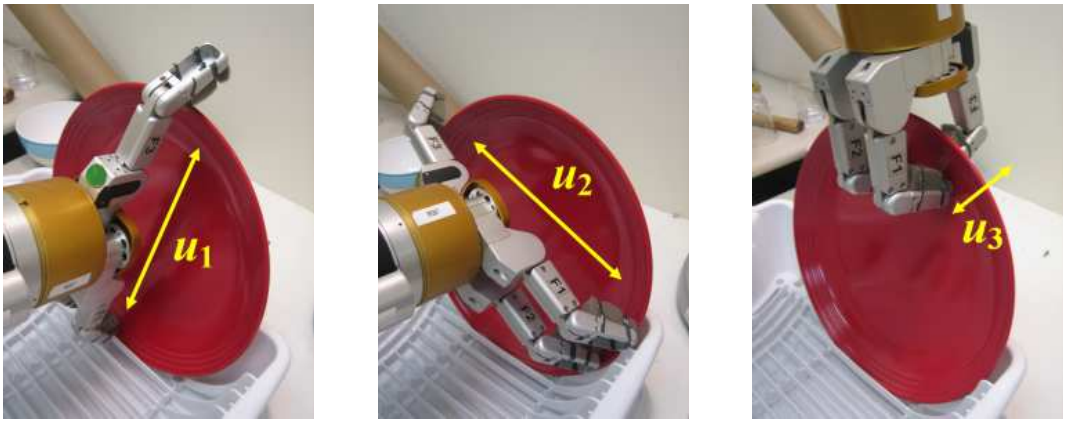


Figure 5.4: Principle components of a plate. h parallel to u_3 is the only good grasp.

5.3.4 Contact Normals

Along the same direction, it is possible to check other orientations. Even if the object's narrow edge is selected by the previous feature, the hand may be tilted, as shown in Figure 5.5. This corresponds to bad yaw and pitch orientations. To prevent tangential slippage forces, the fingers and the palm of the hand should therefore come into contact with the object in the direction of the surface normal. For each finger direction (in which each finger closes) and the out direction (in which the palm reaches toward the grasping point), the direction should be parallel to its corresponding surface normal on the object. The surface normal is the plane of points closest to the fingers/palm along the finger/out directions. From this plane, we can again use SVD to compute the plane normal, and ideally the directional similarity between the finger/out directions and the corresponding surface normal is 1.

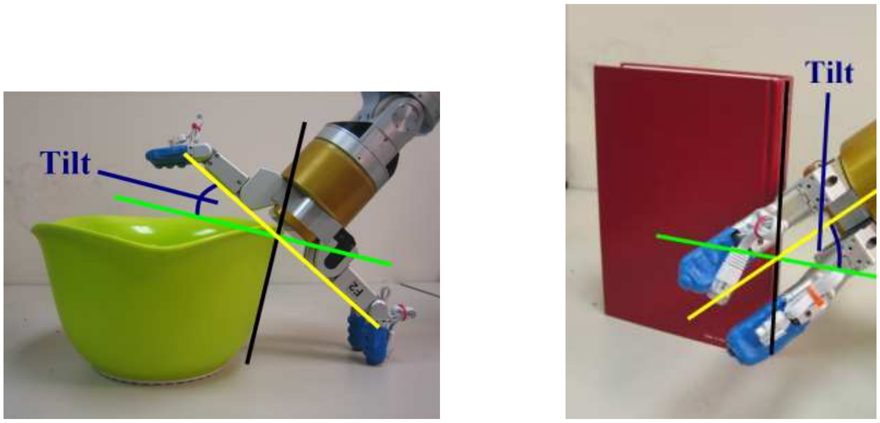


Figure 5.5: Examples of hand tilted even if narrow edge is selected.

5.4 Training

Three classes of objects were considered for training – plates, bowls, and rectangular blocks. For each class, 5 distinct objects of various sizes and shapes were chosen. The SwissRanger 3-D point cloud was collected for these objects. We did not use the Bumblebee2 point cloud for the selection algorithm because it was too incomplete and produced heavy bias toward textured regions such edges, since such regions would then have more points. Grasp configurations for these objects were then collected by manually moving the arm as if to grasp the object. Configurations were then visualized in simulation and labeled 0 (bad grasp) or 1 (good grasp) by hand, based on experience of how successful and unsuccessful grasps appear in simulation. For each object, 10 good and 10 bad grasp configurations were collected to give a balanced training set. Since there were 15 objects, this gave a total of 300 training samples. Features for each of the sample configurations were computed, and we applied logistic regression to learn a logistic model for the relationship between the features and the labels. The model was then stored and used, as in Figure 5.1, to evaluate future grasp configurations.

Chapter 6

Experimental Results

The search and selection algorithms described in the previous two chapters will first be evaluated, then experiments performed on the actual STAIR robot will be discussed.

6.1 Search

As described in Chapter 4, for each of the 20 3-D grasping points generated by our classifiers and for each yaw-pitch parameter pair, the system tries to find a satisfying 6-D inverse kinematics solution. The yaw and pitch spaces are discretized into 7 and 4 steps respectively. For each 6-D solution, 12 different wrist roll configurations are found, and for each 2 spread angles are considered. This gives an 8-D configuration, from which the finger openings are determined. Multiplying the search spaces in each component, for each 3-D grasping point we therefore have at most $7 \times 4 \times 12 \times 2 = 672$ 11-D grasp candidates, and hence in total we have at most $672 \times 20 = 13440$ candidates. While this may appear to be a very large number, most of these candidates are not included in the candidate set, either because an inverse kinematics solution was not found or no finger openings could be found to make the configuration collision-free. In practice, for a given 3-D grasping point only approximately 10% of the search space become included, and since some grasping points are unreachable and give no feasible configurations at all, the total number of candidates is around 1000.

To better understand the role of each component of the search pipeline, we can compare the difference in search performance when components are removed (and the naïve strategy equivalents are used). For each component removal, we ran the search algorithm on the training samples used for the selection algorithm (see Section 5.4), and analyzed the candidate configuration set in simulation. When simply using the naïve method, only one 11-D grasp candidate is found from each 3-D grasping point, so at most 20 grasp candidates can exist; however, this number is usually around 5 only, because the other grasping points did not have inverse kinematics solutions or had configurations in collision with the point cloud. Moreover, at most 1 or 2 were actually good grasps, and in a significant number of cases no good configurations were found at all. The lack in number of good configurations is extremely undesirable, since no matter how well the selection algorithm performs, it cannot produce a configuration that is better than the best one in the candidate set; if the set has no good configurations, then the final grasp must be bad too.

Significant improvement can be seen when the wrist roll search is added. Now instead of one 11-D configuration per 3-D grasping point, 12 configurations are found, giving a candidate set of size around 100. Around 30% of these configurations are good grasps, which at first glance is disappointing because the same percentage held for the naïve method. However, the difference lies in that there are now 30 good grasps in 100 instead of 1-2 good grasps in 5. First, this means that there are many more good grasps for the selection algorithm to choose from, hence eliminating the potential problem in the naïve method where the candidate set had no good grasps at all. Second, not all good grasps are equal; some are better than others. With a greater number of candidates, the likelihood that a better good grasp is produced is higher. Assuming that the selection algorithm can choose a near-optimal or optimal configuration, the final grasp should therefore in general be better if the candidate set was larger. Empirically, the difference is evident; the naïve method succeeds less than 50% of the time, whereas with wrist roll search the success rate is over 70%.

Adding in the wrist direction (yaw-pitch) component did not give as much improvement as expected. The grasp candidate set size was increased by an order of magnitude, but the proportion of them being good grasps dropped slightly. There are two main reasons for the small amount of improvement. First, increasing the number of good grasps from a moderate number to a large number, which is what happens when wrist direction search is added to wrist roll search, is much less significant than increasing the number from close to zero to a moderate number, which is the case when wrist roll is added to the naïve method. The likelihood that a reasonably good grasp exists in the moderate number is high, so increasing the number of good grasps will unlikely produce a much better good grasp, hence adding wrist direction search was much less significant. Second, the system without wrist direction search already contains some degree of sampling in the yaw-pitch space. This exists because inverse kinematics originally had the freedom to choose the orientation of the arm (and hence the wrist), so each of the reachable 3-D grasping points most likely sampled a different part of the yaw-pitch space. Additionally, as many grasping points are very close to each other, this effectively means that some grasping points have multiple configurations with different wrist directions reaching them. This makes the wrist direction search redundant, hence the small improvement increase. Empirically, the success rate increased by approximately 5%.

6.2 Selection

To test the features and see if they are effective, the training samples were split into a training set and a test set, where the logistic classifier was trained on the training set, and this model was tested on the test set. In particular, 10-fold cross validation was used, where the training samples were split into 10 partitions, then 10 tests were performed, each time selecting a different partition as the test set and the other partitions form the training set. Finally, the errors over the 10 tests are averaged. This is a commonly used test.

Of the four categories of features presented in Section 5.3, all were useful except for the contact normals feature. The problem is that in order for surface normals to be accurate, the point cloud region from which a surface normal is found must be small. However, a small region means fewer points, and given the low resolution of the SwissRanger camera, very few points (around 10) end up being considered. The fewer the number of points, the more each one has an influence on SVD (which is used to extract the surface normal from the points), and the less meaningful the SVD is; in fact, the surface normals produced are often inaccurate. The contact normals category of features was therefore excluded and was not used in our empirical experiments discussed in the following sections.

For the other three categories of features, an 85% average test set accuracy was achieved.

6.3 Grasping Single Novel Objects

To test our algorithms on the actual STAIR robot, we first considered grasping single objects in uncluttered environments from 13 novel classes (i.e., of different types from the training set) in a total of 150 experiments. These objects differed significantly in size, shape, and appearance, and are very different from the plates, bowls, and rectangular blocks used in the training set. In each trial, one object was placed randomly on a table in front of the robot. The results of this experiment can be seen in Table 6.1. “Prediction good” refers to the percentage of cases the final grasp returned by the selection algorithm appeared to be a good grasp in simulation, and “actual success” refers to the percentage of cases in which the robot actually grasped the object (and by our definition, was able to pick it up).

We performed 10-20 trials for each object class, and each class had different instances of each object (e.g., multiple apples were used for the “apple” class); this tested the robustness of the algorithm to changes in the object. Our trained classifier was capable of selecting a candidate grasp that succeeded in simulation 81.3% of the time. When these grasps were actually performed, the grasp success rate was 76%. The gap in success rates reflects that the model is imperfect, since grasps that supposedly succeed in the model fail in reality.

We also found that success rate was very different depending on the size of the object, hence size information was also provided in Table 6.1. We defined an object to be small if it could be enclosed within the robot hand, medium if it was approximately 1.5-3 times the size of the hand, and large if the size was greater. Both prediction and actual grasping were best for medium sizes, with success rates of 92% and 86% respectively. On the other hand, success rates were considerably lower for large objects, mainly because the edges where grasps can occur were naturally much thicker for the large objects chosen. This constrained the number of feasible grasps because much fewer grasp configurations could actually fit the edge and be able to apply sufficient force on the object for grasping.

Nevertheless, even though the experimental setup was significantly harder (in terms of the object classes considered) than our previous experiments in [13], our current algorithm was still able to achieve an overall grasp success rate of 76%, which compares to the previous success rate of 70%. Examples of STAIR grasping experiments can be seen in Figure 6.1.

Table 6.1: Grasping Single Objects (150 trials)

Object Class	Object Size	Prediction Good	Actual Success
Ball	Small	80%	80%
Apple	Small	90%	80%
Gamepad	Small	85%	80%
CD container	Small	70%	60%
Helmet	Medium	100%	100%
Ski boot	Medium	80%	80%
Plate bundle	Medium	100%	80%
Box	Medium	90%	90%
Robot arm link	Medium	90%	80%
Cardboard tube	Large	70%	65%
Foam	Large	60%	60%
Styrofoam	Large	80%	70%
Coil of wire	Large	70%	70%

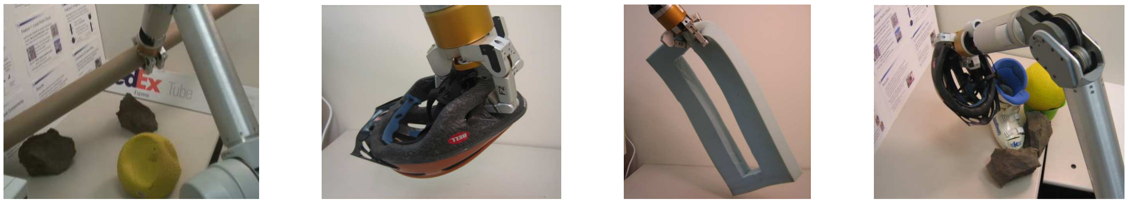


Figure 6.1: Examples of grasping experiments.

6.4 Grasping in Cluttered Environments

The main objective of our algorithm was to address the problem of grasping in cluttered environments. In a total of 40 experiments, each in which more than 5 objects were placed randomly but close to each other (possibly even touching), STAIR was instructed to any single object from the scenario. If any object can be grasped from the scene, the experimental trial was denoted a success. STAIR was able to achieve a success rate of 75% in this task (see Table 6.2), which is surprisingly close to the success rate for single objects, indicating that our algorithm is to some extent robust to clutter.

Table 6.2: Grasping in Cluttered Scenes (40 trials)

Environment	Object Class	Prediction Good	Actual Success
Terrain	Long tube	87.5%	75%
Terrain	Rock	100%	75%
Kitchen	Plate	87.5%	75%
Kitchen	Bowl	75%	75%

6.5 Grasping on a Different Robotic Platform

One property noted when describing the selection algorithm's features in Section 5.3 is that the features are not manipulator-dependent, and can be applied on other robotic hands. To test the validity of this claim, we applied the algorithm on our STAIR 1 robot, the predecessor of the current STAIR robot studied. STAIR 1 parallel-plate gripper whose size is somewhat smaller than the hand discussed so far. In a total of 50 trials of a dishwasher unloading task, where kitchen items were to be removed from a cluttered dishwasher containing 3 or more randomly-placed objects, our algorithm gave results comparable to the previous two tasks (see Table 6.3). Our algorithm can therefore be generalized to different robots.

Table 6.3: Dishwasher Unloading on STAIR 1 (50 trials)

Object Class	Prediction Good	Actual Success
Plate	100%	85%
Bowl	80%	75%
Mug	80%	60%

Chapter 7

Conclusions

To construct a robust grasping system for STAIR, I applied the comparative approach and split the problem into two main subsystems - search and selection. The search algorithm uses input from STAIR's cameras to generate a grasp configuration candidate set, and the selection algorithm ranks each of these candidates and outputs the maximally-scoring one. Combining these two subsystems, we get a system that takes as input camera data of the scene in front of the robot, and gives as output a single grasp configuration (and a grasp trajectory through motion planning) that grasps a target object in the scene.

In search, we used a previously developed image-based classifier to find candidate 3-D grasping points from the camera images and point clouds. A naïve method that simply found a configuration that reached these 3-D grasping points was assessed, and was found to perform poorly. The main problem observed was that arbitrarily choosing the orientation (which was not constrained by the naïve method) often led to bad grasps. To have a chance of getting a good grasp, the space of orientations must therefore be sampled systematically. Hopefully, even if the proportion of good grasps among those found is small, the selection algorithm will be able to choose grasps with correct orientations. Orientations are sampled in two independent spaces, the wrist direction (yaw-pitch) and the wrist roll. The latter sampling proved to give significant improvements empirically, and the former was useful but not as significant. However, due to the brute-force nature of the search, this step is somewhat slow; more intelligent search methods will be considered and developed in the future.

In selection, we trained a logistic classifier based on features of a grasp configuration's local point cloud that were indicative of whether the grasp would succeed/fail. Four categories of features were considered: presence, stability, grasp direction, and contact normals. These correspond to checking the presence of point cloud points within the hand's grasp, the vertical direction distribution of the object (ideally even), the horizontal direction (ideally parallel to small component directions of the region), and the finger/out directions (ideally parallel to the corresponding surface normals). In practice, the surface normal category was not used because there were often too few points on the relevant surface's point cloud, giving uninformative and inaccurate surface normals. This feature is a powerful check for orientation, hence its absence probably gave rise to the empirical observation that many grasps output by the selection algorithm still had suboptimal orientations even with the grasp direction feature. Not having a way to compute surface normals therefore directly affects the performance of the grasping system. The root of this problem is the lack of point cloud points that represent the relevant surface, which is caused by the low resolution of the SwissRanger camera. There appears to be no other way to resolve this issue until a higher resolution camera that gives relatively complete images is available; we are currently developing one that shall be used in the near future.

A more long-term future direction focuses on environment feedback. The grasping system described in this thesis uses input from the environment at one point only – the beginning. From that snapshot of the scene, STAIR is expected to be able to grasp a target object without any other input. For humans, that is analogous to looking at the scene and identifying the target, then blindly moving one's arm toward the object and grasping the object with thick gloves on (hence not actually knowing when one has touched the object). This clearly is a difficult and unreasonable task, because it requires one to be correct from the beginning – there is no margin for error. Instead, when we perform grasping, we rely on continuous visual information to locate the object relative to our hand, and on sensory information from the hand to know when we are touching the object or have collided into something else. Were similar feedback systems implemented on STAIR, it would know how to correct its own errors and stop before doing something bad (e.g., tipping over a bowl of water). We have begun our investigation on force feedback systems, and plan to incorporate them as part of our grasping system in the future.

Nevertheless, through the robotic grasping system presented in this thesis, STAIR is now capable of grasping a wide variety of objects in previously unimaginable environments. This is an important step toward realizing the aim of the STAIR project: to construct an intelligent robotic agent that can facilitate and interact with the general human population.

Acknowledgments

I would like to express sincere thanks toward Ashutosh Saxena and Professor Andrew Ng for providing helpful guidance for this project. Many thanks also to all members of the STAIR Perception-Manipulation team, whose efforts to develop and expand the functionality of the STAIR robots was necessary to make this project possible.

Through the process of working on this undergraduate honors thesis, I have been exposed to and enriched by many new experiences, and I thank the Department of Computer Science of Stanford University for having provided me with such an opportunity. Thanks also for bringing together, through weekly seminars, the small group of CS honors thesis students, whom I got to know better and learn about what projects they work on.

My deepest thanks also goes to my friends, my family, and especially my parents, who have supported and beared with me through this year of hard work. Without your support, I would not be who I am.

Finally, I thank Wylie Wan for her kind, dear, and understanding support.

Bibliography

[1] A. Saxena, L. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *AAAI Conference on Artificial Intelligence*, 2008.

[2] L. Wong, "Learning to select robotic grasps using vision on the Stanford Artificial Intelligence Robot," 2008.

[3] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *International Conference on Robotics and Automation*, 2000, pp. 348–353.

[4] G. Dunn and J. Segen, "Automatic discovery of robotic grasp configurations," in *International Conference on Robotics and Automation*, 1988, pp. 396–401.

[5] J. Pauli, "Learning to recognize and grasp objects," *Machine Learning*, vol. 31, no. 1-3, pp. 239–258, 1998.

[6] A. Anglani, F. Taurisano, R. de Giuseppe, and C. Distante, "Learning to grasp by using visual information," in *Computational Intelligence in Robotics and Automation*, 1999, pp. 7–14.

[7] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Grasping pomdps," in *International Conference on Robotics and Automation*, 2007, pp. 4685–4692.

[8] I. Kamon, T. Flash, and S. Edelman, "Learning to grasp using visual information," in *International Conference on Robotics and Automation*, 1996, pp. 2470–2476.

[9] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An SVM learning approach to robotic grasping," in *International Conference on Robotics and Automation*, 2004, pp. 3512–3518.

[10] E. Chinellato, A. Morales, R. Fisher, and A. del Pobil, "Visual quality measures for characterizing planar robot grasps," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 1, pp. 30–41, 2005.

[11] A. Morales, P. Sanz, A. del Pobil, and A. Fagg, “Vision-based three-finger grasp synthesis constrained by hand geometry,” *Robotics and Autonomous Systems*, vol. 54, no. 6, pp. 496–512, 2006.

[12] A. Saxena, J. Driemeyer, J. Kearns, and A. Ng, “Robotic grasping of novel objects,” in *Neural Information Processing Systems*, 2006, pp. 1209–1216.

[13] A. Saxena, L. Wong, M. Quigley, and A. Ng, “A vision-based system for grasping novel objects in cluttered environments,” in *International Symposium of Robotics Research*, 2007.

[14] A. Saxena, J. Driemeyer, and A. Ng, “Robotic grasping of novel objects using vision,” *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[15] F. Schwarzer, M. Saha, and J.-C. Latombe, “Adaptive dynamic collision checking for single and multiple articulated robots in complex environments,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 338–353, 1995.

[16] L. Wong, “Robot motion planning on STAIR,” 2007.

[17] ——, “Learning to select a good grasp,” 2007.