Learning to Grasp Using Visual Information

Ishay Kamon, Tamar Flash, Shimon Edelman
Dept. of Applied Mathematics and Computer Science
The Weizmann Institute of Science
Rehovot 76100, Israel
Internet: ishay@isaac.cs.technion.ac.il
tamar@wisdom.weizmann.ac.il
edelman@wisdom.weizmann.ac.il

May 18, 1994

Abstract

A scheme for learning to grasp objects using visual information is presented. A system is considered that coordinates a parallel-jaw gripper (hand) and a camera (eye). Given an object, and considering its geometry, the system chooses grasping points, and performs the grasp. The system learns while performing grasping trials. For each grasp we store *location parameters* that code the locations of the grasping points, *quality parameters* that are relevant features for the assessment of grasp quality, and the grade. We learn two separate subproblems: (1) to choose grasping points, and (2) to predict the quality of a given grasp. The location parameters are used to locate grasping points on new target objects. We consider a function from the quality parameters to the grade, learn the function from examples, and later use it to estimate grasp quality. In this way grasp quality for novel situations can be generalized and estimated.

An experimental setup using an AdeptOne manipulator to test this scheme was developed. Given an object, the system takes one image of it with a stationary top-view camera, uses the image to choose two grasping points on the boundary, performs a grasping trial with a parallel-jaw gripper, and assigns a grade to the trial using an additional side-mounted camera. The system has demonstrated an ability to grasp a relatively wide variety of objects, and its performance improves with experience appreciably after a small number of trials.

1 Introduction

A general goal of robotics is to develop systems that gather information through interaction with the world, and to improve their performance based on experience. In the context of grasping, this corresponds to learning to improve the grasp quality. Previous work may be divided into two main approaches:

- The analytic approach takes a model of the target object, and finds optimal grasping points on it, relative to some criteria for optimality (see Nguyen [Ngu88], Markenscoff and Papadimitriu [MP89], Faverjon and Ponce [FP91] and Blake [Bla92]). In the context of learning we examine the function

  $$f1 : S \mapsto A \qquad\qquad s.t.\ G\ is\ maximized$$

  where S = State (sensory information), A = Action, and G = Grade (quality of the action). This formulation is called *reinforcement learning*.

- The comparative approach generates a set of candidate grasps, evaluates the quality of each grasp and chooses the best candidate (see Wolter et al. [WVW85], Gatrell [Gat89], and Francois et al. [FIH91]). The function

  $$f2 : S \times A \mapsto G$$

  is studied.

Relatively a few works have dealt with the problem of learning to grasp. Dunn and Segen [DS88] presented a system that first tried to recognize its target object using a stored library. If the object was recognized, the stored grasp was applied to it. For an unknown object, the system tried to grasp it by trial and error. Tan [Tan90] used a set of features to distinguish among objects. In the training stage, some measurements were taken for several objects, and a decision tree was built, making it possible to distinguish among the objects. In the working stage, a sequence of actions was planned, and the target object was recognized and grasped. No generalization of new objects was performed in those systems.

Salganicoff and Bajcsy [SB92] presented a general framework for learning sensorimotor tasks, considering the function $f2 : S \times A \mapsto G$ and suggesting to approximate it from examples. Given such an approximation and a set of sensory parameters obtained for a new situation, they suggested finding the action parameters expected to give a good grade. In other words, they tried solving the analytic approach problem using the formulation of the comparative approach.

1.1 Our approach

We took the comparative approach, that is, we divided the problem into two subproblems: (1) learning where to grasp, and (2) predicting grasp quality. Where to grasp was learned by storing and applying grasping locations from successful trials. In order to predict grasp quality, we took a version of the function $f2 : S \times A \mapsto G$ from examples. We believe that grasp quality can be predicted rather reliably using a few, mainly local parameters,

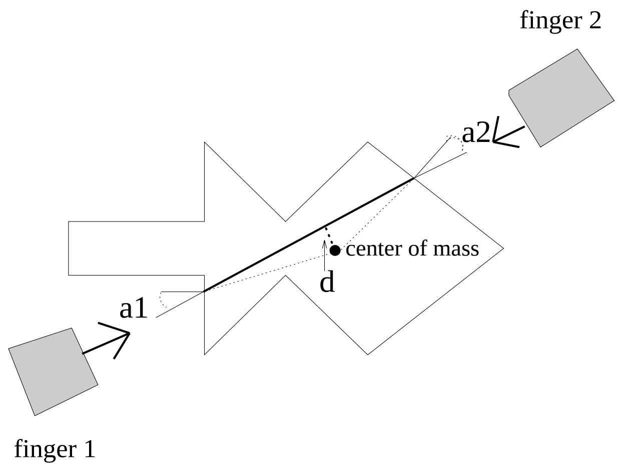


Figure 1: Important features of a grasp configuration: the angles $a1, a2$ between the fingers and the normals at the grasping points, and the distance d between the grasping line and the center of mass.

and consider only parameters that can be extracted visually, for example, the angles between the fingers and the normals at the grasping points, and the distance between the center of mass and the line which connects the grasping points (see figure 1). We found a subset of the most predictive parameters using statistical methods, and called them *quality parameters*. We learn the function $f3 : Q \mapsto G$ from the quality parameters Q to the grade. Given an object and grasping points on it, the quality parameters were extracted, and the learned function to estimate grasp quality used.

The quality parameter space has several advantages for learning. Each point in the parameters space represents a class of grasp configurations that can be used on different objects, which are similar locally, near the grasping points, thus achieving generalization among objects. The number of parameters is small, and does not depend on the complexity of the target objects; therefore a small number of examples is needed for learning. There are explicit relations between the quality parameters and the grade, therefore the mapping is relatively simple and smooth, and generalization among neighboring points in the quality parameter space is possible.

We can rely on a few quality parameters that represent only a partial model of the grasping system because we can observe visually the outcome of the grasping trials. The quality of a grasp is influenced by parameters that are not considered explicitly in our model, such as the force and contact characteristics of the gripper, the dynamic behavior of the soft fingers, etc. The system adjusts to these parameters via the overall quality of the observed grasp. Note that the grasp quality also depends on non visual properties of the target objects (e.g. weight, friction, rigidity etc.). The system adjusts to these parameters by averaging the grades that correspond to the same quality parameters, or by considering the worst case.

2 Preparations

In this section we choose strategies and features for the working system.

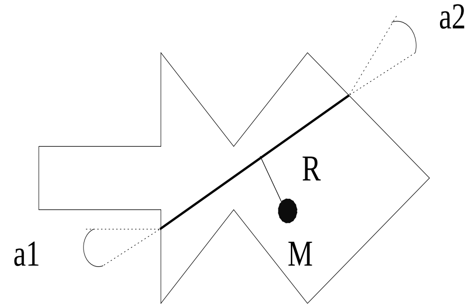


Figure 2: Rotation torque around the grasping line $= R \times M \cdot g$.

Heuristic strategies are used to choose grasping points when stored knowledge from previous trials is not applicable for a novel situation. In this case grasping trials must be performed until a successful grasp occurs. We compared several strategies and showed that using more information about the grasping problem improves performance.

We chose features that can predict the grasp quality. A few features that give the best prediction are chosen from a set of possible features, using statistical methods.

We compared strategies and features using simulation of grasping trials. We used 20 synthetic images obtained by cross-sectioning random generalized-cone objects (see figure 3). Every grasp (two grasping points on an object) gets a grade, assigned by a mechanical model of grasping (see appendix A for details). We calculated the grade, considering two components:

1. No sliding –
   Sliding of the fingers is not allowed. This limits the angles between the line that connects the grasping points (the grasping line) and the normals at the grasping points ($a1, a2$ in figure 2). If these angles are above a threshold, μ, the grade is set to zero.

   If $(a1 > \mu)$ or $(a2 > \mu)$ then $Grade = 0$

2. Resistance to rotation –
   We consider the difference between the resistance to rotation, and the torque that rotates the object around the grasping line.

   The resistance to rotation is the maximal torque the gripper can apply to the object. It depends on the shape and size of the contact areas, the pressure on them and the friction and viscoelasticity of the object and the fingers.

   The torque that operates to rotate the object around the grasping line depends on the mass distribution on both sides of the grasping line. The torque is created by the gravitational acceleration g that operates in the vertical direction. The torque can be represented by the moment arm, R, which operates on the mass of the object, M. (see figure 2).

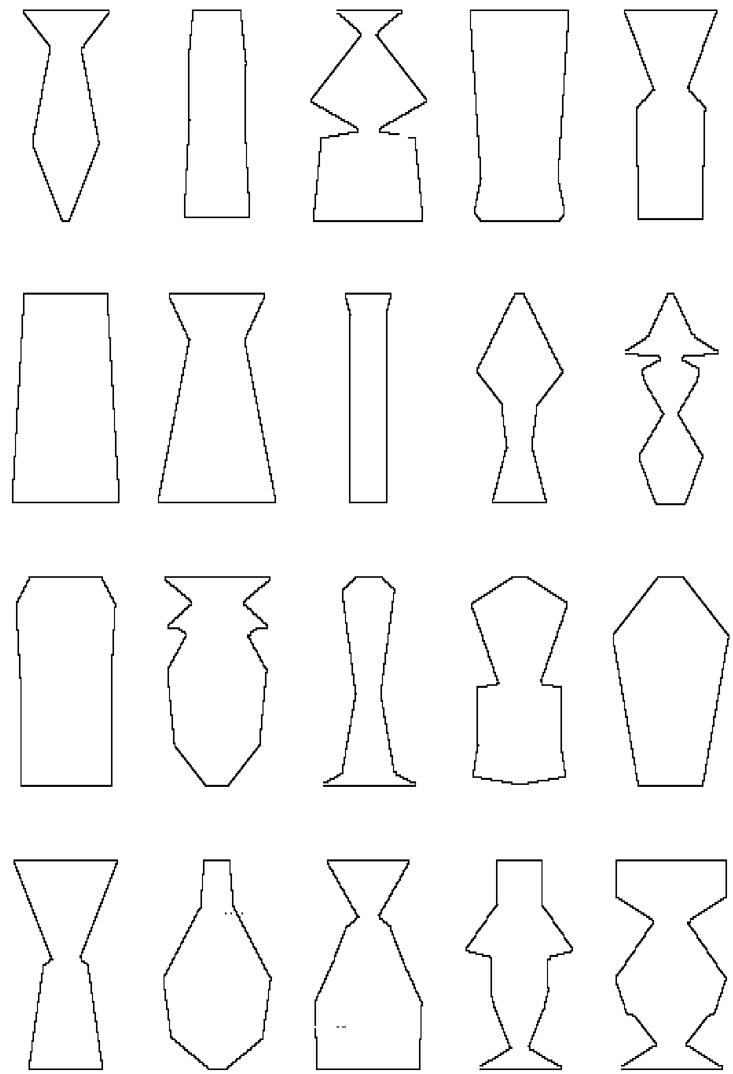


Figure 3: 20 random generalized cones used for simulations.

2.1 Comparing Heuristic Strategies

We compared several strategies and showed that using more information about the grasping problem improves performance. We concluded that the design of a grasping system should include *apriori knowledge*, rather than learning everything from scratch. The five strategies we compared are:

s1 Two points are chosen randomly on the boundary, using a uniform distribution.

s2 The first point is chosen randomly on the boundary, using a uniform distribution. The direction to the second point is chosen relative to the internal normal at the first point, using a zero-mean Gaussian distribution.

s3 The first point is chosen according to its distance from the center of mass, along the main axis. The distance has a zero-mean Gaussian distribution. We randomly choose a distance d, go from the center of mass for a distance d along the main axis, and go perpendicular to the main axis until we reach the boundary. The direction to the second point is chosen relative to the internal normal at the first point, using a zero-mean Gaussian distribution.

s4 Similar to $s3$, but the distributions are tighter. The first point is closer to the center of mass, and the direction to the second point is closer to the internal normal.

s5 Similar to $s4$, but the direction to the second point tends to get closer to the center of mass.

For each heuristic strategy, 2000 grasping trials were simulated. For each object from the 20 target objects, 100 grasp configurations were chosen using the tested heuristic strategy. Every grasp configuration consisted of two grasping points on the boundary of the target object. A grade was calculated for each grasp, assigned by a mechanical model of grasping (see appendix A for details).

strategy	mean	stdv	prob80	prob85	prob90	prob95
s1	8	21	2	2	1	0
s2	42	35	17	11	6	2
s3	53	34	26	18	10	4
s4	67	31	46	35	21	9
s5	70	32	57	45	30	14

Table 1: Comparing heuristic strategies. We show the mean grade and the standard deviation. probxx is the probability to have a grade greater or equal to xx.

The fifth strategy $s5$ gave the best results. The difference between $s4$ and $s5$ was statistically significant (with $p < 0.01$).

The results support the intuitive strategy of grasping on two opposite sides, and near the center of mass.

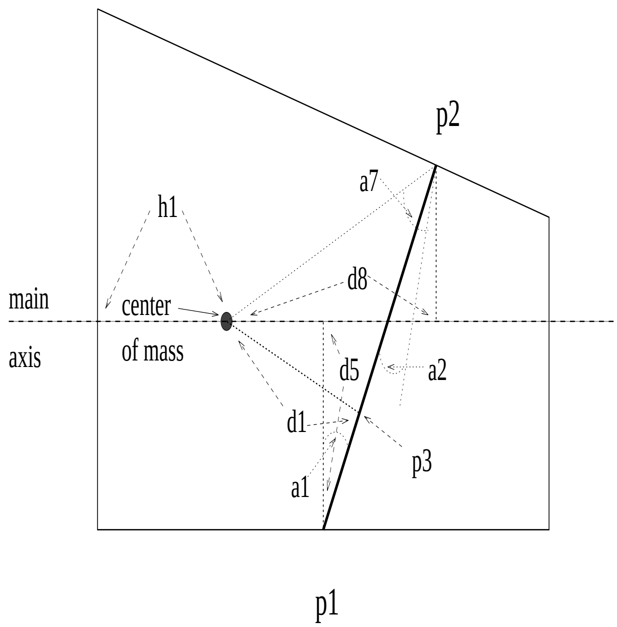


Figure 4: Visual features - important angles and distances.

2.2 Extracting Features

In this section we describe the visual features that can be extracted from an image of a target object. A subset of these features is used to predict the quality of a grasp. In the next section we explain how to choose the most predictive features.

For a given image of a target object we find the main axis and center of mass of the object, and define a coordinate system centered on it. We use the length along the main axis, from the center of mass to the edges, to resolve the ambiguity of the direction of the X axis. The negative direction of the X axis is chosen as the one that has a shorter distance to the edge (see $h1$ in figure 4).

For each grasp configuration consisting of two grasping points on the boundary of the object, we extract 20 features from the image, which are divided into distances and angles. Below are presented a partial list of the most important features.

1. $a1$ - angle from $p1$ to $p2$ (relative to $p1$ internal normal).
2. $a2$ - angle from $p2$ to $p1$ (relative to $p2$ internal normal).
3. $a7$ - angle from $p2$ to the center of mass (relative to $p2$ internal normal).
4. $d1$ - distance from the center of mass to the grasping line.
5. $d5$ - distance from $p1$ to the symmetry axis.
6. $d8$ - projection of $p2$ on the symmetry axis.
7. $d9$ - $d1$ normalized by $h1$.

2.3 Choosing Important Features

A few visual features are necessary to predict the quality of the grasp. In this section we compare features in order to find a small subset of the most predictive ones.

A comparison was made using 2000 configurations of the best strategy $s5$. For each object from the 20 target objects, 100 grasp configurations were chosen. A grade and 20 features were calculated for each grasp. We estimated the statistical relations between the features and the grades (see appendix D for details). The results show that a small subset of features can predict the quality of the grasp rather reliably. The best subset of three parameters consisted of the angles between the fingers and the normals at the grasping points, $a1, a2$, and a normalized distance from the center of mass to the grasping line, $d9$. This triplet gives prediction quality 0.97 according to the conditional average prediction method. These features are considered in the intuitive strategy for two-fingered grasping, that is, grasp on opposite sides, near the center of mass. They are also the most important attributes of a grasp configuration, considering the mechanics of the grasping problem (see appendix A).

The above results support the statement that a few visual features are sufficient to predict the quality of a grasp. We next show how to use this information in the design of a robotic system that learns to grasp.

3 The Working System

3.1 Overview

The working system consists of four components: the control subsystem, the learning subsystem, the vision subsystem and the action subsystem.

The user first presents an object in the field of view of a stationary top-view camera, and initiates a grasping trial. A picture is taken and processed by the vision subsystem. The result is a segmented image of the target object, its center of mass, and the direction of the main axis. The learning subsystem chooses grasping points on the object's boundary, using stored information from previous trials. The points in the image coordinates are transformed into action parameters for the robot, and the action subsystem performs the grasping trial. The grasped object is presented in front of a second, side-mounted camera, which takes a picture of the object grasped by the gripper. The vision subsystem processes this image and calculates the quality of the grasp. If the quality is good enough, the new example is stored by the learning subsystem, to be used in the future. The control subsystem coordinates the operation of the other subsystems.

The software for the control and the learning subsystems is written in lisp, and runs on a Sun4 machine. The vision subsystem uses two fixed cameras that are connected to a SGI Indigo 4000 machine. The image processing program is written in C, and runs on the SGI. The action subsystem consists of the AdeptOne robot with a parallel-jaw gripper.

Communication between the Sun4 and the SGI is performed by running remote programs (using the Unix rsh command), and writing to common files, using a TCP/IP network, and The communication between the Sun4 and the Adept robot is by using an interface program that runs on a different Sun4, which has a serial line connection to the robot.

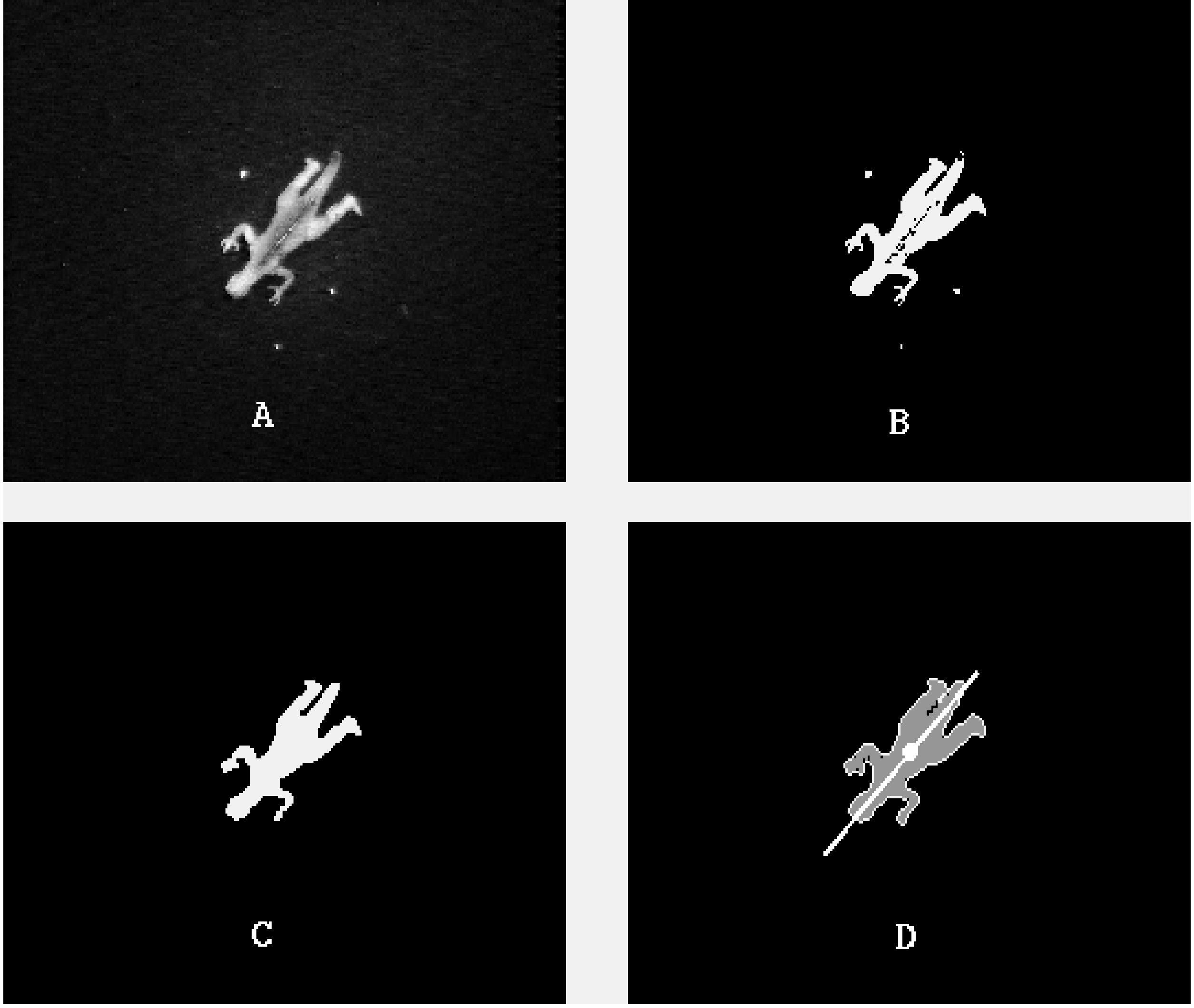


Figure 5: Stages in image processing of a toy dinosaur image – (A) A gray level image; (B) Thresholding the image; (C) Filling holes and removing small patches; (D) Marking the boundary, center of mass and main axis.

3.2 The Vision Subsystem

The vision subsystem takes pictures with a camera, using the SGI svideo library. We use 576×768 8-bit gray level images. The result of the image processing is a segmented image of the target object, with its center of mass, and the direction of the main axis. The interior and the boundary of the object are marked separately.

3.2.1 Image Processing

The gray-level image is segmented, using a fixed threshold. Small object patches (smaller than 50 pixels) are removed, to reduce noise. Background holes inside the object are filled, and considered parts of the object. The object shape is smoothed, using local considerations. The planar center of mass is found by averaging the (x, y) coordinates of the object pixels. The axis of the least second moment is found, by using the method described in Horn [Hor86, page 53]. The boundary of the object is marked forming a closed curve. We mark off pixels in the boundary that are not necessary for connectivity. After thinning, every pixel in the boundary has exactly two neighbors.

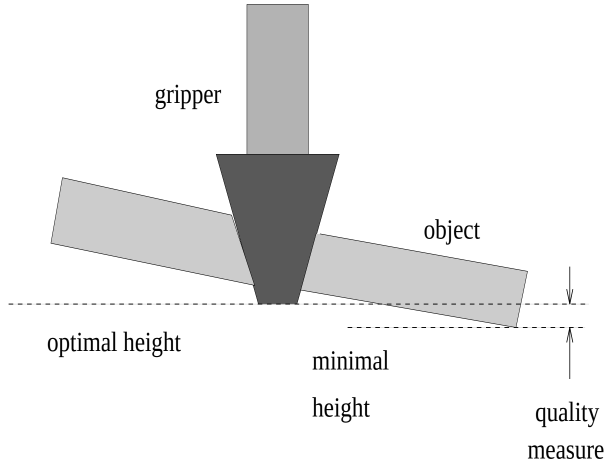


Figure 6: Automatic grading – a side view of an object held in the gripper. The grade (quality measure) depends on the difference between the minimal height and the optimal height.

3.2.2 Automatic Grading

To estimate the quality of a grasp, we used the minimal height of the object (see [DS88]). If the object is held firmly, its height is above a certain value. If the object slides, its minimal height is lower. The grade given is lower when the minimal height is lower (see figure 6).

A more detailed analysis of the object pose in the gripper was not applied because the initial pose of the object was not known. The fixed side camera cannot get a side picture of the object in its initial position, and therefore we do not have a baseline for comparison, which is needed in order to judge whether the object has moved relative to the gripper during the grasping action.

3.3 The Control Subsystem

The control subsystem includes the user interface, coordinates the operation of the other subsystems, and is also responsible for several other tasks.

The learning subsystem needs visual information in order to choose the grasping points. It uses two primitives:

1. Locate a point on the boundary.
   Given a point $p1$ inside the object, and a direction $dir1$, go from $p1$ in direction $dir1$ until reaching the boundary at point $p2$.

2. Given a point $p2$ on the boundary, calculate the orientation of the normal there.
   If the curvature near $p2$ is too big, no orientation is calculated. Otherwise two neighboring points on the boundary are found, and the orientation of the line that connects them is calculated.

The control subsystem also verifies that the open gripper does not collide with the object before reaching the grasping points and transforms the grasping points into action

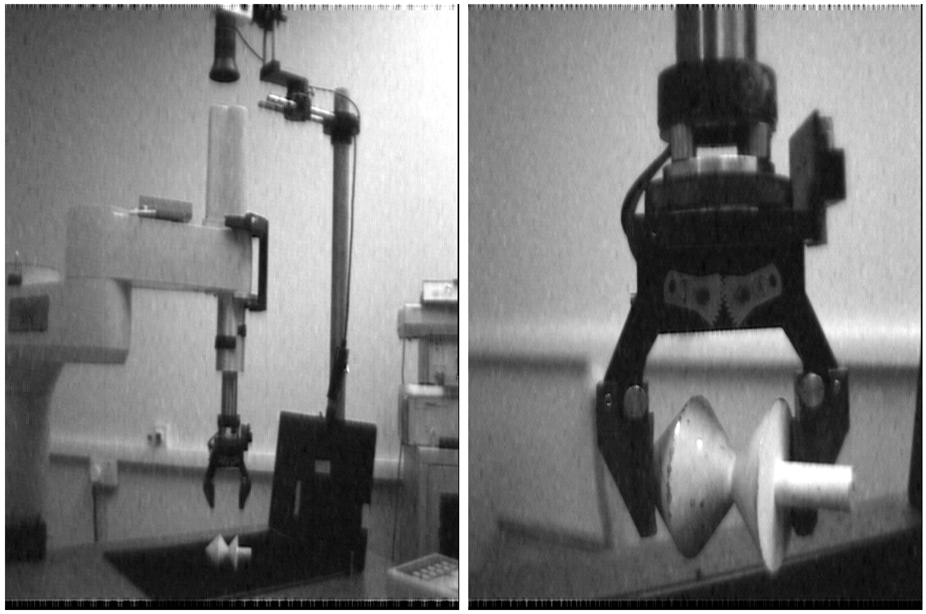


Figure 7: Left: the working environment – the Adept robot with a parallel-jaw gripper, the fixed top-view camera, the black background of the work area and a white target object. Right: the gripper holds the object.

parameters for the robot (see appendices B,C for details).

3.4 The Action Subsystem

The action subsystem consists of the AdeptOne robot with a parallel-jaw gripper (see figure 7). The robot is controlled by the Val-II operating system, and has four degrees of freedom. It can reach any (x, y, z) location in its workspace, with rotational angle θ around the vertical axis. The gripper was built especially for our project. It has only two configurations: open and closed. The force exerted by the fingers is fixed, and the fingers can be covered with various pads to increase friction and compliance. The application program that runs under VAL-II reads commands from a serial port that is connected to an interface program on a Sun4. The possible commands are *move*, that moves the gripper to a specified (x, y, z, θ) location in the robot coordinates, and *open* and *close*, that control the parallel-jaw gripper.

The initial location of the parallel-jaw gripper is out of the field of view of the top-view camera. Given a triple (x, y, θ) as the target location of the gripper, it first moves to $(x, y, z - approach, \theta)$, that is located above the target position. The gripper moves high enough to avoid collision with the target object. It then moves down to a fixed grasping height $(x, y, z - grasp, \theta)$, and closes the fingers. After grasping the object, the gripper takes it in front of the side-mounted camera, in order to estimate the grasp quality. Finally the object is returned to its initial position, and the gripper goes out of the field of view of the top-view camera.

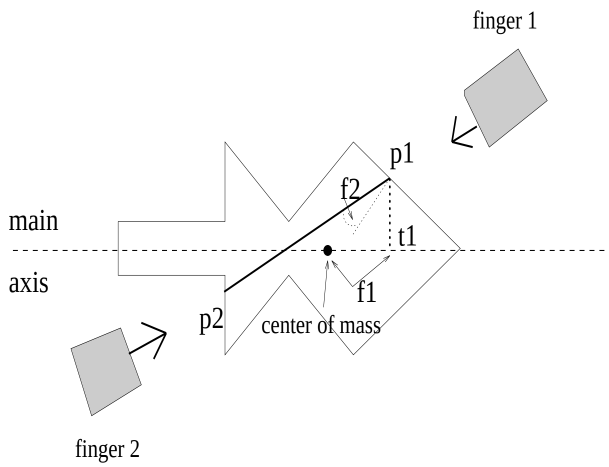


Figure 8: A Heuristic strategy for choosing grasping points, using location parameters $f1, f2$. Go from the center of mass for a distance $f1$. Go in a perpendicular direction to the main axis, until you reach the boundary at point $p1$. Go in the direction $f2$ to the internal normal direction, until you reach the boundary at point $p2$.

3.5 A Heuristic Strategy

A heuristic strategy is used to choose grasping points when stored knowledge from previous trials is not applicable for a novel situation. The heuristic strategy we used was denoted in section 2.1 as $s5$. Given a segmented image, we traverse a distance $f1$ from the center of mass along the main axis to the point $t1$ (see figure 8). The distance $f1$ was chosen randomly, using a zero-mean Gaussian distribution. We go from $t1$ along a perpendicular direction to the main axis, until reaching the boundary at point $p1$. This is chosen to be the first grasping point. We calculate the boundary orientation at $p1$, and go in the direction $f2$ relative to the internal normal until reaching the boundary at point $p2$, which is chosen to be the second grasping point. The angle $f2$ is chosen randomly, using a Gaussian distribution. The mean of the Gaussian is slightly shifted from the internal normal at $p1$, toward the center of mass of the object.

Boundary orientations must be defined at $p1, p2$, that is, it must be guaranteed that the curvature of the boundary at those points is low.

3.6 Coding a Grasp

Grasp configurations are coded as tuples of numbers. The parameters $f1, f2$ that were chosen randomly by the heuristic strategy are used to locate grasping points on the object's boundary. We call them *location parameters*. We also store a few *quality parameters* that are used to estimate the grasp quality. The angles between the fingers and the normals at the grasping points $a1, a2$, and a normalized distance from the center of mass to the grasping line $d9$ are used (see figure 9). These parameters are chosen as the most predictive set of three parameters, as discussed in section 2.3. Note that $f2 = a1$, that is the same parameter is used to locate the second grasping point, and also to estimate the grasp quality.

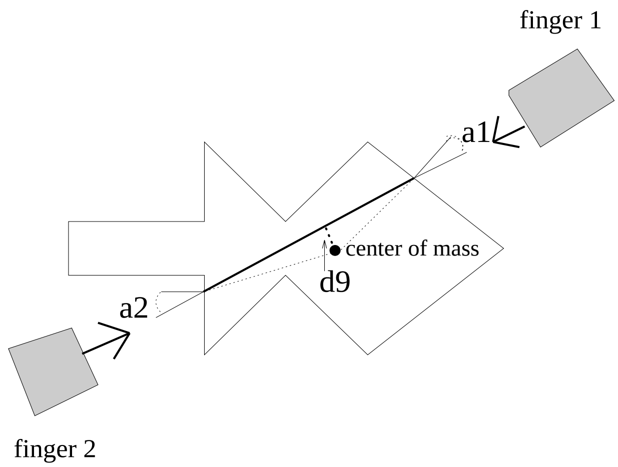


Figure 9: Depicted are the quality parameters of a grasp configuration – the angles $a1, a2$ between the fingers and the normals at the grasping points, and the distance $d9$ between the grasping line and the center of mass.

3.7 The Learning Subsystem

In this section we present the learning mechanism. We distinguish between learning where to grasp, which is coded by the location parameters of stored grasps, and learning to estimate the grasp quality, which is coded by the quality parameters. The latter can be generalized and used to predict grasp quality in novel situations.

The working system presented below learns while performing grasping trials. There is no distinction between the training and working stages. The system starts only with heuristic knowledge which is used to choose grasping points. If a successful grasp is performed, its grasp configuration is stored for future use. Information from the stored grasps is used in three ways:

1. Where to grasp.
   Location parameters are used to locate the grasping points on new target objects.

2. Local estimation of the grasp quality.
   Each set of quality parameters from one grasp is considered a point in a parameter space, and is assigned a grade. Considering the set of measurements from the new image as a new point, we look for its nearest neighbor. We assume that if the new point is close enough to a stored example, its quality should be similar to that of the stored example. Note that we abandon the link between the location of the grasping points on a specific object and the quality parameters that characterize the grasp quality. In this way we can generalize novel grasp configurations.

3. Global estimation of grasp quality.
   We use the stored grasps to define ranges of acceptable parameter values that predict good quality. Grasps that have measurements out of these ranges are ruled out.

Given a new target object, and having a list of stored grasp configurations, we try to apply the stored grasps to the new object.

For each stored grasp, we locate two grasping points on the object's boundary, using the location parameters. We calculate the quality parameters for the new object, and try to match them to the quality parameters of all the stored examples. If a match is found, the grasp is performed, and its grade is expected to be similar to the grade of the matched example.

If no match is found for all stored grasps, we use the heuristic strategy to create grasping points, that is, we calculate the quality parameters for the new object, and try to match them to all the stored examples. Before performing a grasping trial, the parameters are checked to be within the acceptable ranges of the parameter values.

If the last two tests fail several times in a row, the grasp that was chosen by the heuristics is used without considering the knowledge already stored in the system.

3.7.1 Nearest Neighbor Mechanism

Given a new point in the quality parameter space, we look for its nearest neighbor in the list of stored examples. The example list is scanned, and checked for each example e whether the new point is contained within a 3-dimensional box centered at e; and if it is, the new point matches the example e, and the grade corresponding to the example e is the expectation for the new point.

The search is performed in three iterations, increasing the size of the neighborhood box from 3 to 6 to 9 (the units are angles and normalized distances).

Note that similarity is defined according to the lack of difference among the quality parameters, that is objects are similar locally, near the grasping points. For every stored grasp there is a class of grasps on different objects that are similar to it.

3.7.2 Maintaining Stored Grasps

Successful grasp configurations are stored in a common data structure (a list). The considerations in the storage mechanism reflect the problems of the real system. In particular, there is the problem that the quality parameters do not determine the quality of the grasp completely, and therefore similar points in the parameter space may have very different grades. Some grasp configurations are *canonical*, that is they can be applied to many objects, whereas other grasps are *specific* to the objects for which they were used. The distinction between canonical and specific examples applies both to the quality parameters that characterize the grasp quality, and to the location parameters that code the location of the grasping points.
For example, $a1 = a2 = d9 = 0$ and $f1 = f2 = 0$ are examples of canonical sets of quality and location parameters, respectively. The quality parameters refer to grasping an object on two parallel faces, where the grasping line passes through the center of mass. The location parameters add the information that the grasping line is perpendicular to the main axis.

For each grasp configuration the following details are stored: location parameters, quality parameters, average grade and number of matched trials (the grade over all trials that match that point in the quality parameters space is averaged). The neighborhood size is also stored - we can limit the size of the matching neighborhood, based on contradictory examples.

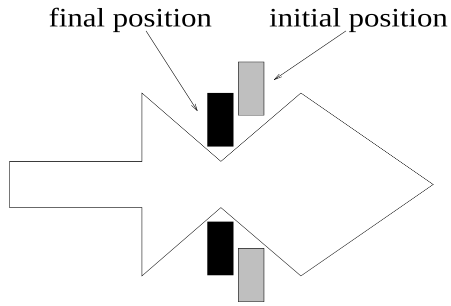


Figure 10: A specific grasp configuration – the fingers slip from the unstable initial position to the final stable position. In the final position, each finger contacts the object at two points that have different orientations. This situation, however, is not considered in our model.

Grasp configurations that have a grade ≥ 95 are stored. The order of the stored grasps affects both the order of attempting grasp locations on the target object, and matching examples in the quality parameters space. The more canonical examples are to be used first. The stored grasps are sorted according to their average grade (primary key) and the number of matched trials (secondary key).

If a stored grasp is matched with unsuccessful trials, it should be taken out of the list. This situation may happen for specific grasps. The main reasons for differences in grasp quality for the same parameters are:

1. Weight difference.
   Grasps of light objects are less sensitive to the distance from the center of mass. Applying such grasps to heavier objects may result in failure. Note that grasps of heavy objects are applicable to lighter objects.

2. Specific geometry considerations.
   Grasps may rely on specific geometrical details of the grasped object. For example, the fingers may slip from the initial unstable position into a stable grasp (see figure 10). In other cases each finger has contact with more than one point on the object boundary, and therefore the model used for the grasp quality is not sufficient (see final position in figure 10).

3. Marginal grasps.
   Marginal grasp configurations may cause different grades for very similar situations. The difference may result from small differences in the image measurements, small differences by the gripper location relative to the object, contact of one finger before the other finger, etc.

A grasp is removed if its average grade is lower than 90, or the current matched trial grade is lower than 70.

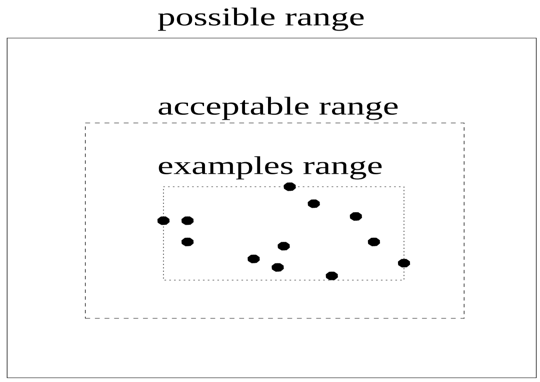


Figure 11: A 2D display of parameter ranges – the internal range of the stored examples, the external range of possible values, and the intermediate acceptable range.

3.7.3 Parameter Ranges

The stored grasps are used to define ranges of acceptable parameter values that would predict good quality. Grasps that have measurements out of those ranges are ruled out. In the beginning the acceptable ranges cover the whole parameters space, and they shrink as more examples are gathered.

For each quality parameter we determine the minimal and maximal values of the stored examples, and then define the acceptable range by adding margins to these values (see figure 11). The size of the margins is proportional to the difference between the minimal (maximal) value and the possible minimum (maximum) value.

$$Range\ min = Example\ min - Portion \times (Example\ min - Minimum)$$

$$Range\ max = Example\ max + Portion \times (Maximum - Example\ max)$$

The portion of the margin size depends on the number of good trials the system has performed. It decreases as the number of examples increases, using the formula :

$$Portion = exp(-1 * \frac{number\ of\ trials}{10})$$

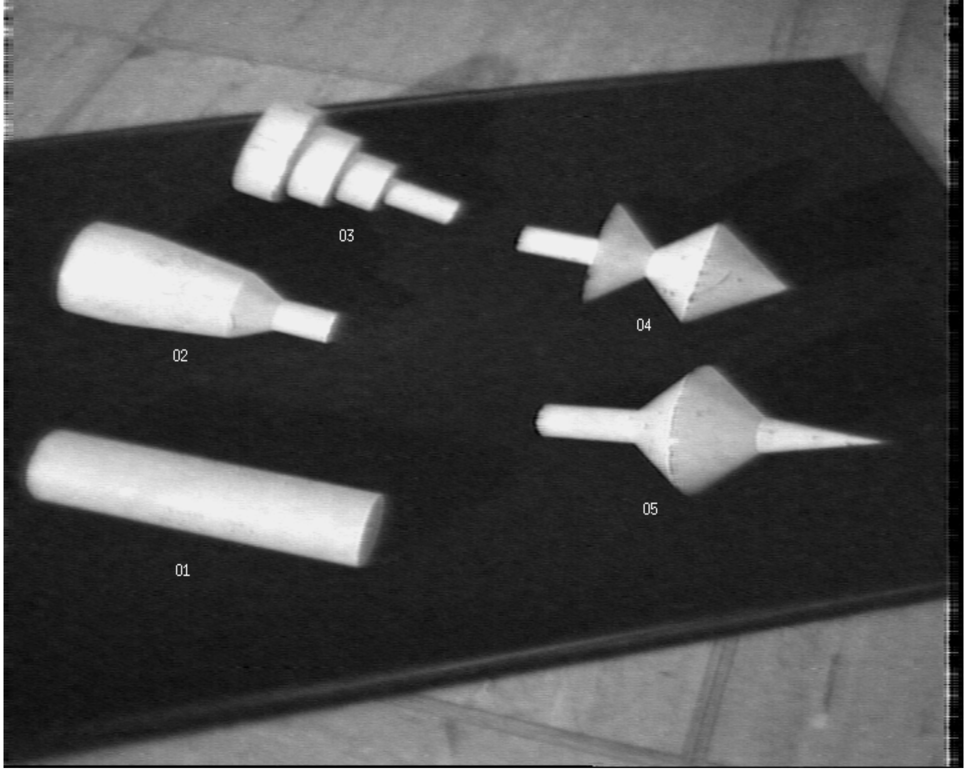


Figure 12: The target objects $o1, o2, o3, o4, o5$

4 Experimental Results

Two experiments were performed to test the above scheme. In each experiment a series of objects was repeatally presented to the system (one object at a time). The objects were positioned by hand near the center of the work area. The improvement in performance was measured as the system stores information about successful trials.

4.1 Experiment 1

In this experiment generalized cones were used as target objects. At each iteration there was one trial for every object, in the order of $o5, o3, o4, o1, o2$ (the order was randomly chosen). For each iteration the average grade, the minimal grade, the success rate (percentage of $grade \geq 80$), and the number of examples were measured.

The experiment consisted of three sessions, each starting with no previously stored grasps - the first two sessions, of 20 iterations (100 grasping trials) each, and the third session, of 25 iterations (125 grasping trials). A larger number of trials were done because the system needed more time to stabilize on a high level of performance. Each trial took about 1 minute.

In order to compare the results to non-learning trials, 10 iterations (50 grasping trials) were performed using only the heuristic strategy, and the same characteristics (average and minimal grade, success rate) were measured for each iteration. The average of the those results was considered a base line for comparison. The following graphs present the experimental results.

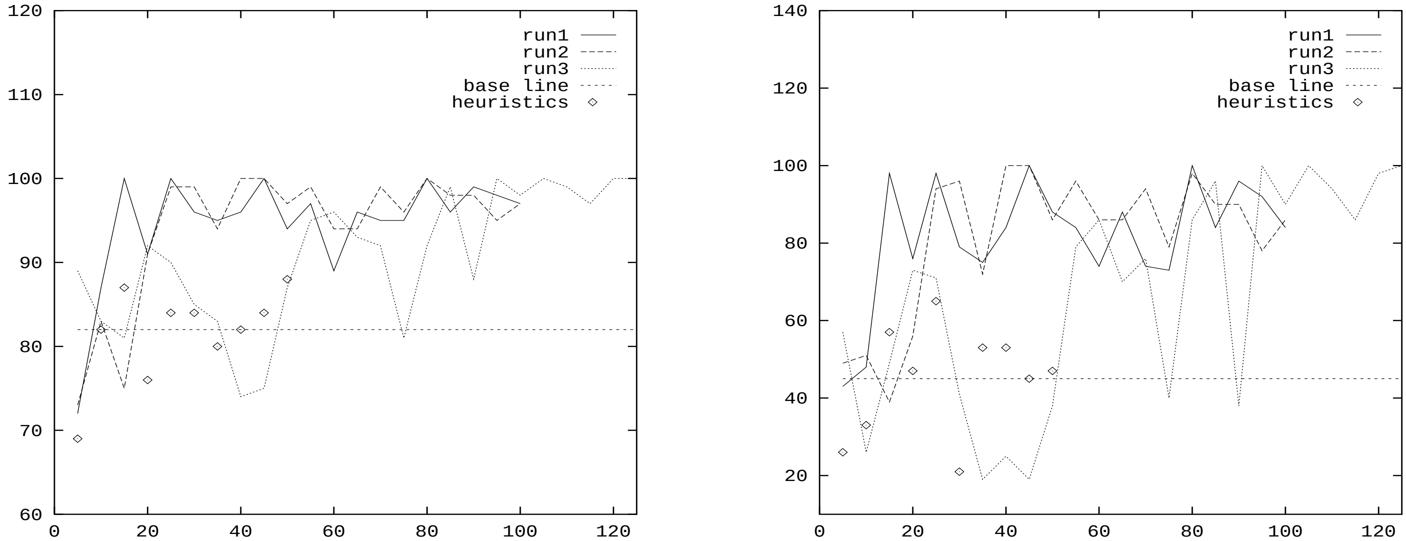


Figure 13: Experiment 1 – average (left) and minimal (right) grades. The X axis represents the number of trials, and the Y axis shows the grades. Each run started with only heuristic knowledge. The average (minimal) grades increased as the system gained experience. Every point shows the average (minimal) grade for a single iteration over five grasping trials. The base line is the average of 10 iterations of trials using only the heuristic strategy.

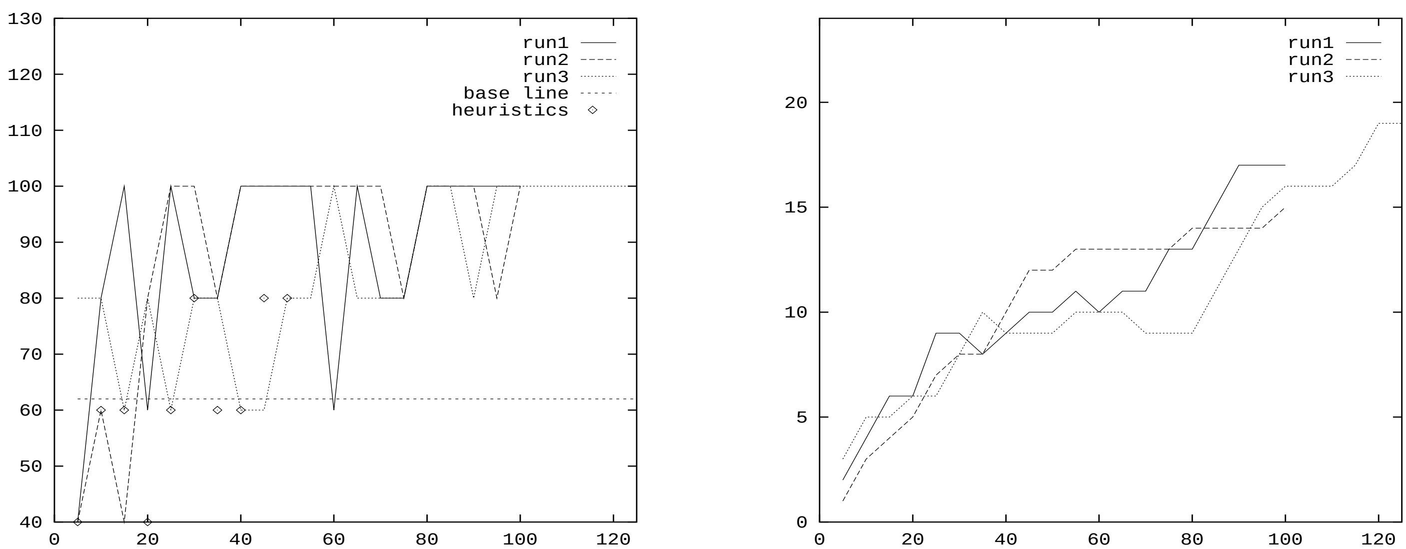


Figure 14: Experiment 1 – success rate and number of examples. Left: success rate – the X axis represents the number of trials. The Y axis shows the grades. Every point shows the success rate ($grade \geq 80$) for a single iteration over five grasping trials. Right: number of examples – the X axis represents the number of trials. The Y axis shows the number of stored examples.

4.2 Experiment 2

In this experiment 15 target objects where used, which varied in size, weight, rigidity and color (see figures 15 and 16), and included the five generalized cones, three stones, a cup, a soda can, a tennis ball, a plastic plug, and three plastic toys (a doll, a hammer, and a dinosaur).

The weights of the target objects ranged from 830g for the heaviest stone, to 30g for the soda can. The objects were presented in a random order. At each iteration we performed one trial for every object. and measured the average grade, the minimal grade, the success rate (percentage of $grade \geq 80$), and the number of examples.

We performed one session starting with no stored grasps, consisting of 15 iterations (225 grasping trials).

In order to compare these results to non-learning trials, we performed 5 iterations (75 grasping trials) using only the heuristic strategy, and measured the same characteristics (average and minimal grade, success rate) for each iteration. The average of those results was considered a base line for comparison.

Grasp configurations that were learned for certain objects were applied to other objects. The system successfully tolerated instability of the vision subsystem (the most unstable feature was the direction of the main axis). New examples were stored even after 200 trials, which suggests that the system did not completely stabilize. However, the number of new examples added, decreased as the system gained more experience (see graph of number of stored examples, figure 18).

The ranges of parameters that successfully predicted good grasp quality are presented below, and include the ranges that were learned in experiment 2 using 52 stored examples, after 199 successful grasping trials. The ranges for features $a1, a2$ refer to angles. Small angles between the fingers and the boundary normals at the grasping points were found to be necessary for good grasps, as was a small distance from the center of mass. The range for $d9$ refers to the percentage of normalized distances.

feature	minimal value	maximal value
a1	-20	17
a2	-12	24
d9	-20	14

Table 2: Acceptable ranges of parameters. We present the ranges that were learned in experiment 2.

The base line for the average grade of experiment 2 is high relative to the average grade of experiment 1, because many of the target objects in experiment 2 were of light, and therefore easy to grasp. A few objects were difficult to grasp and caused failures. After completing the experiment described above, the system successfully grasped new objects, for which it did not have previous experience, which confirmed the ability to generalize to new objects.

The following pictures and graphs present the target objects and examples of grasping configurations, and the experimental results, respectively.

Figure 15: The target objects – left: three stones, a toy dinosaur, and a tennis ball. Right: A doll, a plug, a soda can, a cup and a plastic hammer.

Figure 16: The gripper holds the doll (left) and the object o3 (right).

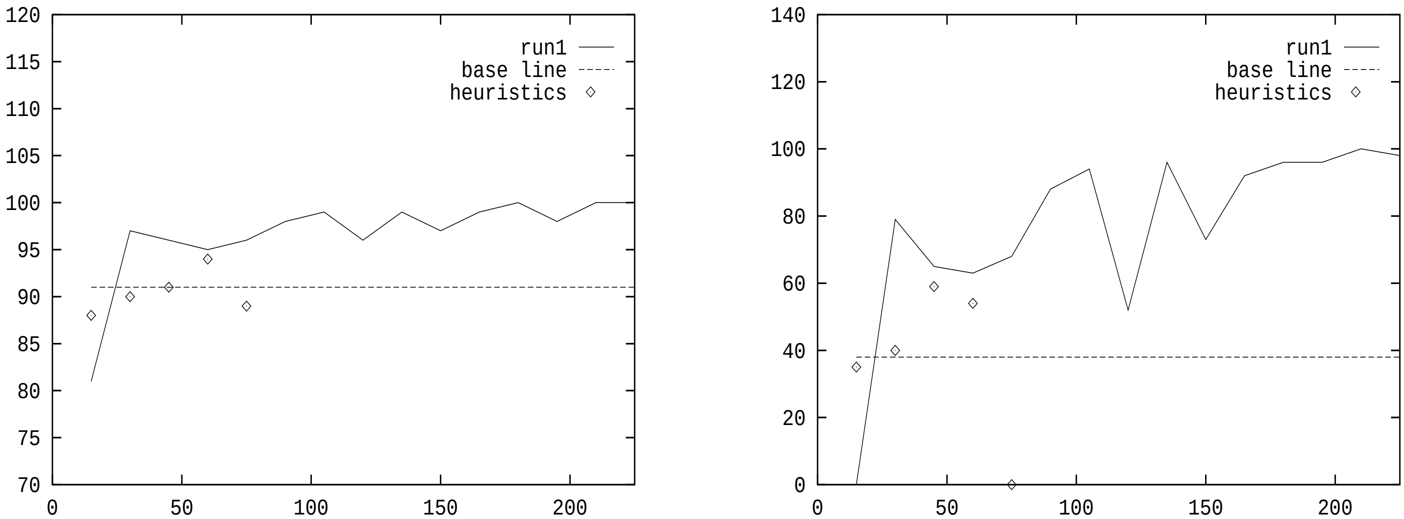


Figure 17: Experiment 2 – average (left) and minimal (right) grades. The X axis represents the number of trials, and the Y axis shows the grades. The run started with only heuristic knowledge. The average (minimal) grade increased as the system gained experience. Every point shows the average (minimal) grade for a single iteration over 15 grasping trials. The base line is the average of 5 iterations of trials using only the heuristic strategy.

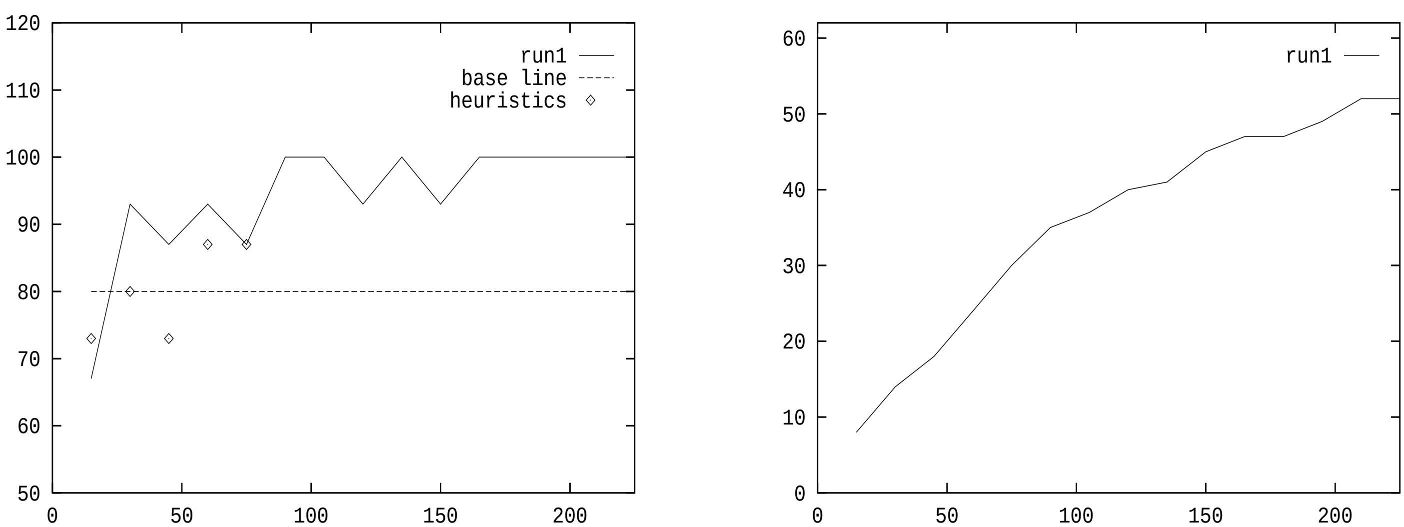


Figure 18: Experiment 2 – success rate and number of examples. Left: success rate – the X axis represents the number of trials. The Y axis shows the grades. Every point shows the success rate ($grade \geq 80$) for a single iteration over 15 grasping trials. Right: number of examples – the X axis represents the number of trials. The Y axis shows the number of stored examples.

5 Discussion

5.1 Task Difficulty

In this section we utilized the experience of hundreds of grasping trials observations to judge the difficulty of the visually guided grasping task.

Our first observation was that grasping light objects, using soft fingers, is quite easy. *Light* is defined relative to the characteristics of the gripper – the force applied by the fingers, the friction and softness of the fingers, etc. An object is considered light if the grasp quality does not depend on the location of the center of mass, relative to the locations of the grasping points. Our results indicated that in this case, it is sufficient to choose grasping points that keep the angles between the fingers and the contact surfaces small (below a certain slippage threshold).

As the objects became heavier, the shape and size of the contact areas and the location of the center of mass were found to become more important. The difficulty of the task increased, depending on two factors: geometric considerations and uncertainty about the target object's characteristics.

1. Geometric considerations.
   The most *canonical* grasp configuration for the type of objects we used (filled, elongated, with straight axis) is grasping near the center of mass, on two opposite surfaces, where the grasping line is perpendicular to the main axis. For certain shapes, this grasp was not applicable (because of corners near the center of mass, or non existence of parallel surfaces, for example, in the case of a triangle). Finding good grasps for this kind of shapes is difficult. There are several algorithms that solve the problem of finding stable two-fingered grasp for 2D objects (see Nguyen [Ngu88], Markenscoff and Papadimitriu [MP89], Faverjon and Ponce [FP91] and Blake [Bla92]). These algorithms require the shape of the target object, which is represented as a polygon or a smooth curve, and perform non trivial geometrical analysis. There are objects that do not have a stable two-fingered grasp.

2. Uncertainty about the object's characteristics.
   A single image of the target object cannot supply all the information necessary for grasping. In particular the location of the center of mass, the shape and size of the contact areas can only be estimated. This uncertainty increases the probability of failure, and makes it necessary to perform active explorations (i.e. try to grasp the object, and assign a grade to the grasp based on the observed results).

5.2 Advantages

The main advantages of the scheme presented here are:

- Generalization.
  Knowledge from previous trials can be generalized to novel objects. This is related mainly to learning canonical grasp locations, learning to predict grasp quality for a given configuration, and learning ranges of parameters that predict good grasps.

- Learning of specific grasps.
  In addition to general knowledge, the proposed scheme can learn specific grasps that are suitable for specific objects. This is possible because the scheme stores locations of good grasping points.

- Simple computation.

  1. Simple image processing - no recovery of shape.
  2. Compact representations - grasp coding requires a relatively small amount of memory. Matching and other manipulations are very simple.
  3. Simple control of the robot - only position control is used. The gripper is passive, and does not require force control, force feedback, etc.

- Small number of trials.
  An appreciable improvement of performance occurs after a small number of grasping trials.

- Modularity.
  The scheme consists of four components: the control, vision, learning and action subsystems, which communicate through simple interfaces. The subsystems deliver parameters to each other, but conceptually there is no shared data structure. The implemented system runs on four different machines.

5.3 Comparing Learning to A Mechanical Model

The predictions of the learning system suggested are compared to the quality measure calculated by the mechanical model described in section 2.

We used 2000 grasp configurations of the 20 random objects described in section 2. For each object, 100 grasps were chosen by the heuristic strategy $s5$. The relevant features for grasp quality $a1, a2, d9$ were calculated from the images, as described in section 2.2.

For each grasp, a quality measure was calculated by the learning system, using the learned knowledge from experiment 2, as described in section 4.2. The learned knowledge consists of ranges of parameters that predict good grasps, and a list of stored grasps, that are considered as points in the quality parameters space.

For each grasp we first checked if the features $a1, a2, d9$ were within the acceptable ranges. If they were not, the grasp was labeled "not acceptable". If they were, we tried to match them with the stored grasps. If a match was found, the corresponding grade of the matched example was assigned to the grasp. If no match was found, the grasp was labeled "acceptable". To conclude, a quality measure from the learning system can be either a number (a match is found), or an "acceptable" / "not acceptable" label. For each grasp, a numerical grade is assigned by the mechanical model.

The following table contains the comparison results. At each line a range of grades assigned by the mechanical model is presented, percentage of grasps that were accepted/not accepted by the range check, and percentage of grasps that match stored examples with differences of 5, 10, and 15 between the learning grade and the model grade.

range of grades	not acceptable	acceptable	diff 5	diff 10	diff 15
0 - 60	100	0	0	0	0
61 - 80	80	20	0	0	1
81 - 90	41	59	2	21	23
91 - 95	39	61	40	16	0
96 - 100	21	79	73	0	0

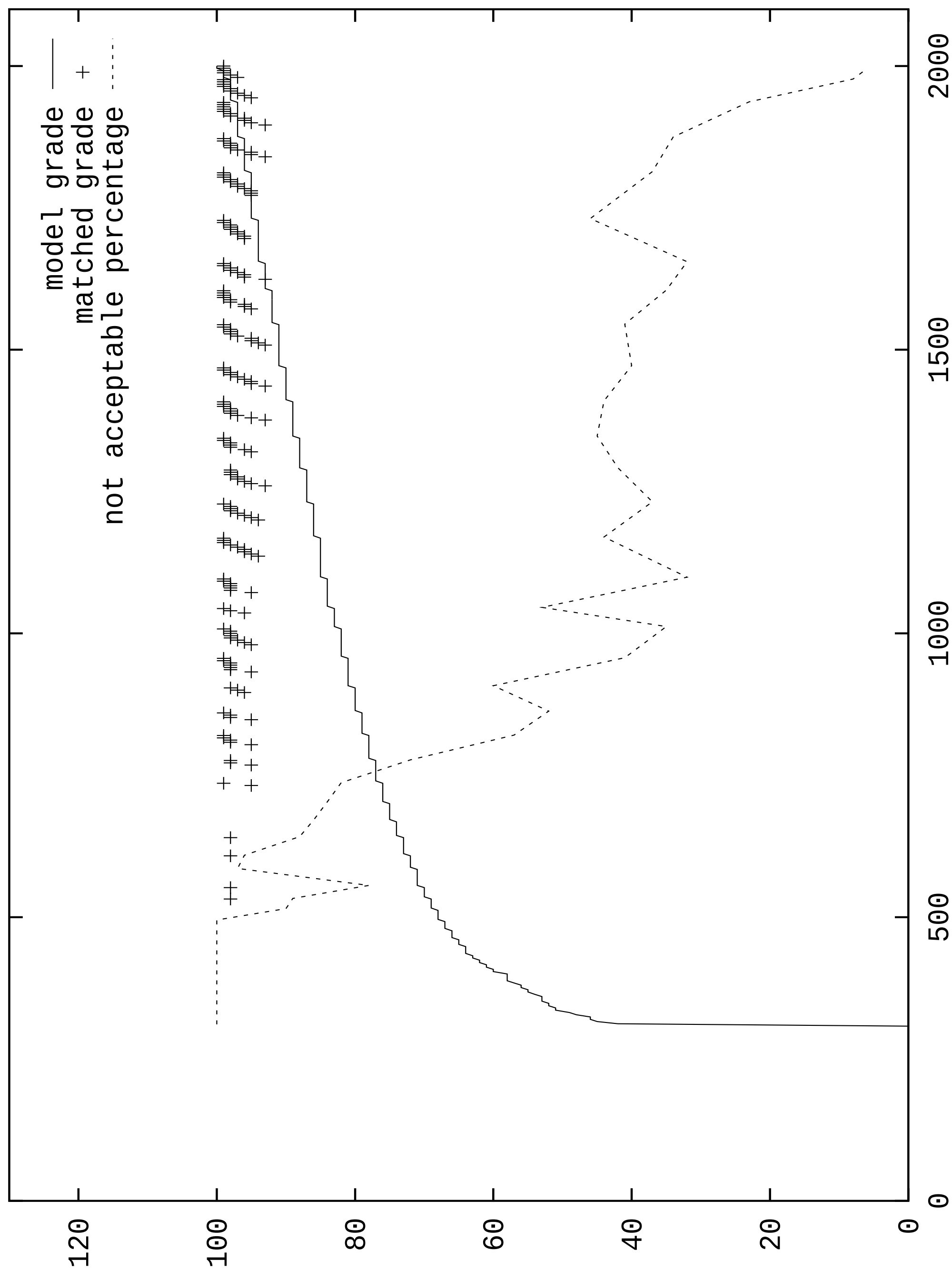
Table 3: Comparing learning to a mechanical model. At each line a range of grades assigned by the mechanical model is presented, percentage of grasps that were accepted/not accepted by the range check, and percentage of grasps that match stored examples with differences of 5, 10, and 15 between the learning grade and the model grade.

- All the grasps with very low grade, $0 - 60$, were ruled out by the range check.
- 80 percent of the grasps with low grade, $61 - 80$, were ruled out by the range check.
- The portion of grasps ruled out by the range test decreases as the grade increases.
- The portion of grasps that closely matches the stored examples increases as the grade increases.

The learning system usually agrees with the mechanical model concerning very successful and unsuccessful grasps. There is a difference for the intermediate level of grades (80-95). Note that the mechanical model was not tuned to the physical characteristics of the real gripper, and therefore, not more accurate than the learning system.

The following graph presents the results of the above comparison. A histogram of the model grades is showen by the solid line. The X axis represents the number of trials, and the Y axis shows the grades. If a match was found for a trial, its grade is represented as a dot. The number of matches increases as the grade value increases.

Another plot that is superimposed on the above graph presents the percentage of the trials that were ruled out by the range check. One hundred percent of the trials that received the grade zero were ruled out, and that portion decreased as grades increased.


model grade
matched grade
not acceptable percentage
0
500
1000
1500
2000
0
20
40
60
80
100
120

5.4 Design of a Grasping System

We present a design for a grasping system that incorporates the experience and knowledge learned while working with the system described in this work.

- Embedded knowledge.
  The system should have apriori knowledge for estimating grasp quality, which may be in the form of ranges of parameters that predict very good grasp quality.

- Structured search for a grasp.
  The system should start the search for grasping points by trying the canonical grasp locations. It will use the embedded knowledge to predict if the grasp locations are good. If the canonical grasps are not applicable, the system will look for grasping points in a structured way, e.g. starting from configurations where the grasping line passes trough the center of mass, and gradually increasing the distance from the center of mass.

- Storing successful grasps.
  The system should store information about successful grasps in ways similar to the scheme presented in this work. The learned information will extend the apriori knowledge and adjust it to the properties of the actual system.

- Object recognition/classification.
  For objects that require a specific grasp, it seems natural to attach the grasp configuration to an internal representation of the object and to apply the grasp if the new object matches this representation. The level of representation may vary from a complete reconstruction of the target object, to local descriptions near the grasping points. This requires additional mechanisms for creating such representations, and for matching the new object with stored examples. Such additional information will resolve the problem of contradiction between examples, because grasp configurations of different quality should be represented differently.

5.5 Summary

We presented a scheme for learning visually guided grasping, and a robotic system that implemented and tested the scheme. Our system successfully learned to grasp a large variety of objects, with very different characteristics (geometry, weight, rigidity, color). The system showed an appreciable improvement of performance after a small number of trials, and maintained a high level of performance over sessions of several dozens of trials. The system generalized among objects. These results have demonstrated better performance compared to previous studies that have dealt with the same problem: Dunn and Segen [DS88] and Tan [Tan90] used three target objects each, and Salganicoff [Sal92] considered only cylinders and boxes. None of the previous studies presented an improvement of grasping performance over a continuous session of work.

The results support the argument that a single image may be sufficient for grasping a 3D object, under reasonable assumptions. It also supports the claim that the quality of the grasp can be predicted using a few visual features that contain local information about the grasping areas and their relations to the center of mass. Recovery of the target object's shape is not necessary.

The lessons from this work may be applicable to learning sensorimotor tasks in general, and especially for learning visually guided tasks. We consider a system that consists of two modules, the first generating candidate actions, and the second estimating their quality. Both modules work in an alternating fashion until an action that is expected to provide satisfactory performance is generated. The system then performs this action. The module that generates actions may combine heuristic knowledge with good stored examples. This type of knowledge is domain-specific, and currently we do not have a general framework for learning it. Prediction of action quality can be formulated as a function from a few parameters to the quality, which is learned from examples and can be generalized.

The learning becomes easier as the number of parameters decreases, therefore only minimal and necessary information should be used. We suggest choosing a subset of the most predictive parameters, using statistical methods. The relevant information depends on the task, therefore specific task-depended representations can be used. This argument supports the purposive and task-oriented approach in sensory information processing, in contrast to the reconstruction approach that calls for a unified and global representation used for all purposes.

The vision subsystem is active in the sense that it performs calculations on demand, in contrast to a one-shot computation. It first calculates global features of the target objects, such as center of mass and direction of the main axis. It then performs local calculations, such as boundary tracing and orientation determination. These calculations are performed on very low level data – the segmented image, and the boundary. Therefore, a higher level representation and shape descriptors are not necessary.

Appendix A – Evaluating Grasp Quality: A Mechanical Model

We assigned grades to the simulated grasping trials using a mechanical model of the grasping operation. Given an image that contains a cross-section of a generalized cone and two grasping points on the boundary, we calculated the grade, considering two components:

1. No sliding –
   Sliding of the fingers is not allowed. This limits the angles between the line that connects the grasping points (the grasping line) and the normals at the grasping points ($a1, a2$ in figure 19). If these angles are above a threshold, μ, the grade is set to zero. If $(a1 > \mu)$ or $(a2 > \mu)$ then $Grade = 0$

2. Resistance to rotation –
   We consider the difference between the resistance to rotation, and the torque that rotates the object around the grasping line.

The resistance to rotation is the maximal torque the gripper can apply to the object. It depends on the shape and size of the contact areas, the pressure on them, and the friction and viscoelasticity of the object and the fingers. We consider soft-finger contacts, and assume that the contact areas are small, therefore each contact can be modeled considering

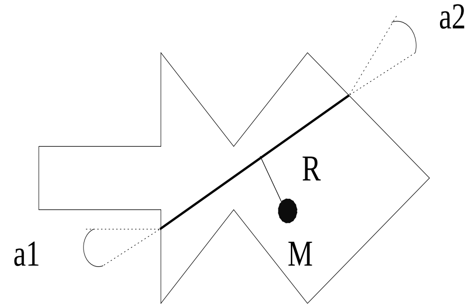


Figure 19: Rotation torque around the grasping line $= R \times M \cdot g$.

only the normal, at one point of contact. It is assumed that the contact areas are flat and have a constant area. Therefore, the contact characteristics are the same for all the grasps. We assume that the grasping force applied by each finger is constant, and its direction is along the grasping line. The two fingers apply forces in opposite directions. The pressure of a finger on a contact area depends on the normal component of the grasping force. The resistance at each contact area is given by the torque τ_i, where K represents the geometrical characteristics of the contact area, μ_s is the coefficient of static friction, and $Force$ is the magnitude of force applied by each finger. The product $K \cdot \mu_s \cdot Force$ is constant over all grasps.

$$\tau_i = K \cdot \mu_s \cdot Force \cdot \cos a_i \qquad i = 1, 2$$

The maximal torque the gripper can apply to the object is the sum of the maximal torques at the contact areas.

$$Resistance = \tau_1 + \tau_2$$

The torque that operates to rotate the object around the grasping line depends on the mass distribution on both sides of the grasping line. We assume that the images are cross-sections of generalized cones, and that the mass distribution per unit volume is constant. The volume is calculated assuming a circular cross-section in depth, perpendicular to the symmetry axis. The torque around the grasping line is given by the formula

$$Torque = (\int_V r \cdot \rho dV) \times g = R \times M \cdot g$$

We integrate over the volume V, where ρ is the density, and r is the moment arm of a mass element ρdV. The gravitational acceleration, g, operates in the vertical direction. The equivalent moment arm, R, operates on the mass of the object, M (see figure 19).

To calculate the grade we consider the *net torque*, which is the difference between the resistance and rotation torque.

$$Diff = Best - min(Best, Resistance - Torque)$$

$$Grade = 100 \times exp(-\frac{Diff}{sigma})$$

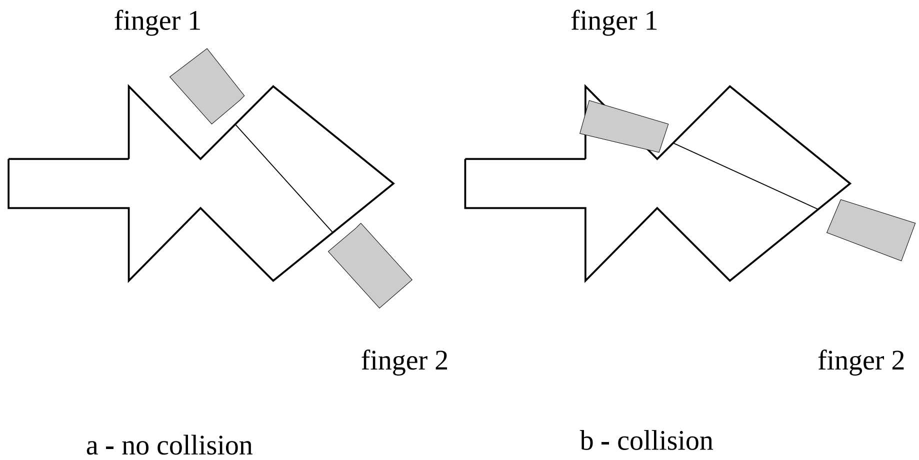


Figure 20: Safety check – (a) A safe grasp and (b) A collision between fingers and object.

The value of $Best$ is the minimal net torque that gives the grade 100. The grade is 100 for $Normal - Torque \geq Best$, and goes down as $Torque$ gets bigger, or $Resistance$ gets smaller. The quality of the grasp decreases exponentially with the net torque (there is no special importance to the exponential relation).

Appendix B – Safety Checks

Given two grasping points on the image boundary, we verify that the open gripper does not collide with the object before reaching the grasping points (see [DS88]). We use two tests :

1. The distance between the grasping points does not exceed the opening of the gripper.
2. The fingers do not collide with the object. The space the fingers use between their initial open positions to the final configuration is projected into the image, and we check that it does not intersect with the object (see figure 20).

Appendix C – Transformation into Action Parameters

Given two grasping points on the object's boundary, we transform them into action parameters for the robot.

First we transform the two points from the image coordinates into the robot coordinates. This process is termed *calibration* between the camera and the robot. We assume a planar mapping, that is the grasping points in the image correspond to points that lie at the same horizontal plane, in the real world. To calculate the transformation, we use four points as examples, and find their locations in both image and robot coordinates. We then solve a system of equations, as described in Ballard and Brown [BB82, pages 481–484]. The calibration was performed once. The system worked for several months using the same transformation parameters.

The two grasping points in robot coordinates are then transformed into a triple (x, y, θ), where x, y is the middle point between the grasping points, and θ is the direction of the grasping line. This triple is passed as action parameters to the action subsystem.

Appendix D – Choosing Important Features

A few visual features are necessary to predict the quality of the grasp. In this section we compare features in order to find a small subset of the most predictive features.

We made the comparison using 2000 configurations of the best strategy $s5$. For each object from the 20 target objects, 100 grasp configurations were chosen. Every grasp configuration consists of two grasping points, on the boundary of the target object. A grade and 20 features were calculated for each grasp. We estimated the statistical relations between the features and the grades.

5.6 Multivariate Regression

Multivariate regression was used to estimate the importance of the features. The regression finds the coefficients for a linear combination of parameters that fits the data best. A model of 60 parameters was considered, where each of the 20 features appeared as x, x^2, x^3. This enabled the regression to consider nonlinear relations between the parameters (features) and the depended variable (the grade).

$$Y = c_1 \times a1 + c_2 \times a1^2 + c_3 \times a1^3 + c_4 \times a2 + c_5 \times a2^2 + c_6 \times a2^3 ....$$

The REG procedure of SAS was used, with MAXR option. It uses forward selection to fit the best one-variable model, the best two-variable model, and so on. Variables are switched so that R^2 is maximized.

The correlation for the best model was $R^2 = 0.82$, using 57 parameters. This means that the data were not approximated very well by the linear combinations of the first three powers for each feature. However small subsets of parameters gave high correlations, relative to the result of the full model. The following table presents the accumulated R^2 of the best subsets of one to nine variables. At each line the added parameter and R^2 of the subset are shown.

Parameter	Accumulated R^2	Percent of full model
$a2^2$	0.633	77
$d1$	0.715	87
$a7^2$	0.735	90
$a2^3$	0.746	91
$a1^2$	0.751	92

Table 4: Importance of features – multivariate regression. The first line shows the best one-variable model, the second line the best two-variable model, etc. The left column shows the parameter added to the subset at each stage.

Running a model with only $a1, a2, d1$ (three powers for each feature) gave a correlation $R^2 = 0.748$, that is 91 percent of the full model.

5.7 Conditional Average Prediction

Because the correlation found by the multivariate regression was not high, we also tested the non-parametric measure of conditional average prediction. In this method a subset of parameters is considered at each stage. The observations (grasp configurations) are divided into bins. Each bin contains configurations that have certain ranges of values of the considered parameters. The expected grade for each bin is the average grade of the configurations in it. The prediction error is the ratio of the variance in the bins SSE and the overall variance SST.

For each subset of parameters we calculated the prediction quality

$$Prediction = 1 - \frac{SSE}{SST}$$

We calculated the quality measure for every pair of features.
The following table presents the pairs with the best prediction quality. Several pairs have prediction quality higher than 0.9. The average prediction quality over all pairs was 0.443.

Feature 1	Feature 2	Prediction quality
$a2$	$d9$	0.95
$a2$	$a5$	0.94
$a2$	$d8$	0.94
$a2$	$d1$	0.93

Table 5: Importance of features – conditional average prediction. The best pairs of parameters are presented with their prediction quality.

We also calculated the prediction quality for several triplets. Adding the third feature $a1$ to the best pair $a2, d9$ gave the best improvement - prediction quality 0.97.

References

[BB82] D. Ballard and C. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersy, 1982.

[Bla92] A. Blake. Computational modelling of hand-eye coordination. *Phil. Trans. R. Soc. London*, 337:351–360, 1992.

[DS88] G. Dunn and J. Segen. Automatic discovery of robotic grasp configurations. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 396–401, April 1988.

[FIH91] C. Francois, K. Ikeuchi, and M. Herbert. A three-finger gripper for manipulation in unstructured environments. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 2261–2266, 1991.

[FP91] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps on curved 2d objects. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 424–429, April 1991.

[Gat89] L. Gatrell. Cad-based grasp synthesis utilizing polygons,edges and vertexes. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 184–189, 1989.

[Hor86] B. Horn. *Robot Vision.* MIT Press, 1986.

[MP89] X. Markenscoff and C.H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics research*, 8(2):17–29, 1989.

[Ngu88] V. D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics research*, 7(3):3–16, June 1988.

[Sal92] M. Salganicoff. *Learning and Forgetting for Perception-Action : A Projection Pursuit and Density Adaptivr Approach.* PhD thesis, University of Pennsylvania, School of Engineering and Applied Sciencs, Computer and Information Science Department, 1992.

[SB92] M. Salganicoff and Ruzena K. Bajcsy. Robotic sensory learning in continuous domains. Proceedings of the IEEE Conference on Robotics and Automation, may 1992.

[Tan90] M. Tan. CSL: A cost sensitive learning system for sensing and grasping objects. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 858–863, May 1990.

[WVW85] J.D. Wolter, R.A. Volz, and A.C. Woo. Automatic generation of gripping positions. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(2):204–213, april 1985.