Automatic Grasp Planning Using Shape Primitives

Andrew T. Miller[†]   Steffen Knoop[‡]   Henrik I. Christensen[*]   Peter K. Allen[†]

[†]Dept. of Computer Science, Columbia University, New York, NY
[‡]IAIM, University of Karlsruhe, Karlsruhe, Germany
[*]Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, Sweden

Abstract

Automatic grasp planning for robotic hands is a difficult problem because of the huge number of possible hand configurations. However, humans simplify the problem by choosing an appropriate prehensile posture appropriate for the object and task to be performed. By modeling an object as a set of shape primitives, such as spheres, cylinders, cones and boxes, we can use a set of rules to generate a set of grasp starting positions and pregrasp shapes that can then be tested on the object model. Each grasp is tested and evaluated within our grasping simulator "GraspIt!", and the best grasps are presented to the user. The simulator can also plan grasps in a complex environment involving obstacles and the reachability constraints of a robot arm.

1 Introduction

Selecting a good grasp of an object using an articulated robotic hand is a difficult problem because of the huge number of possibilities. Even for a simple three-fingered hand such as the Barrett Hand, there are a total of 10 degrees of freedom: 6 degrees of freedom in placing the wrist relative to the object and 4 internal degrees of freedom which set the finger positions. More complex hands have even more possibilities. Of course, large portions of this 10 dimensional space are worthless because the fingers are not in contact with the object, but even if the problem were reparameterized, a brute force search would still be intractable.

A variety of other approaches have been used to tackle this problem. A number of papers present contact-level grasp synthesis algorithms [8, 13, 10, 3]. These algorithms are concerned only with finding a fixed number of contact locations without regard to hand geometry. Other systems built for use with a particular hand restrict the problem to choosing precision fingertip grasps, where there is only one contact per finger [1, 5]. These types of grasps are good for manipulating an object, but are not necessarily the most stable grasps because they do not use inner finger surfaces or the palm. Pollard developed a method of adapting a given prototype grasp of one object to another object, such that the quality of the new grasp would be at least 75% of the quality of the original one [12]. However, even this process required a parallel algorithm running on supercomputer to be computed efficiently.

One way of limiting the large number of possible hand configurations is to use grasp preshapes. Before grasping an object, humans unconsciously simplify the task to selecting one of only a few different prehensile postures appropriate for the object and for the task to be performed. Medical literature has attempted to classify these postures into grasp taxonomies, and the most well known of which was put forth by Napier [11]. Cutkosky and Wright [2] extended his classification to the types of grips needed in a manufacturing environment and examined how the task and object geometry affect the choice of grasp. Iberall [6] reviewed many of the previous grasp taxonomies and generalized them using the concept of virtual fingers. Stansfield [14] chose a simple classification and built a rule based system that, when given a simplified object description from a vision subsystem, will provide a set of possible hand preshapes and reach directions for the pre-contact stage of grasping. However, the system could not evaluate the completed grasps, and thus could not differentiate between them.

In our own work, we have created a grasping simulator, called "GraspIt!", which we have used for analyzing and visualizing the grasps of a variety of different hands and objects[1] [9]. Recently we have expanded the system so that we can automatically plan stable grasps of an object. This planner consists of two parts, one to generate a set of starting grasp locations based on a simplified object model, and one to test the feasibility and evaluate the quality of these grasps. The simplified object model consists of a small set of shape primitives such as spheres, cylinders, cones and boxes, and heuristic grasping strategies for these shapes allow the system to generate a set of grasp possibilities that

---

[1]The complete system will soon be available for download for a variety of platforms from http://www.cs.columbia.edu/~amiller/graspit.
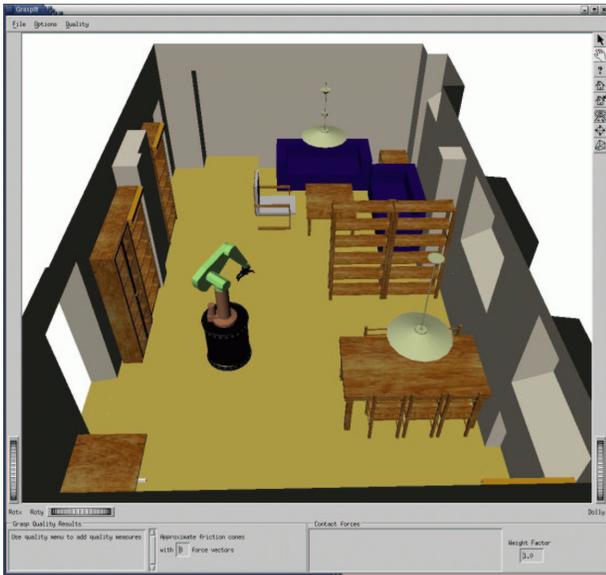
Figure 1: The GraspIt! system allows the importation of a robotic platform and model of the world in which it operates. In this case it is the manipulation platform and living room environment at the Centre for Autonomous Systems. The furniture serves as obstacles in the grasp planning.

are most likely to result in high quality grasps of the object. The grasp tester moves the hand from a grasp starting position toward the object, closes the fingers around the object, and evaluates the grasp. After testing all of the generated grasp possibilities, the user is presented with the best grasps of the object in descending order of quality. In order to prevent infeasible grasps from being planned, the user may import a world model containing obstacles, as well as a robot arm model so that reachability constraints may be considered.

Our goal is to ultimately use this system to plan the grasping tasks of a service robot operating within a home environment (see figure 1). We have shown that with the aid of a vision system it is possible to rectify the poses of elements within the simulated world with their counterparts in the real world, and after a grasp has been planned, it can be executed accurately by the real robot [7].

The paper is laid out as follows. First, we provide a brief overview of the functionality of GraspIt!. Then in section 3 we describe the hand we are using and its possible pregrasp postures. Next, we outline the rules used to generate the set of grasps to be tested. Section 5 describes how each of these candidate grasps is tested and evaluated. Section 6 presents the results of planning grasps for different objects in both an isolated environment and in the presence of obstacles, and finally in section 7 we discuss ways in which the system can be extended.
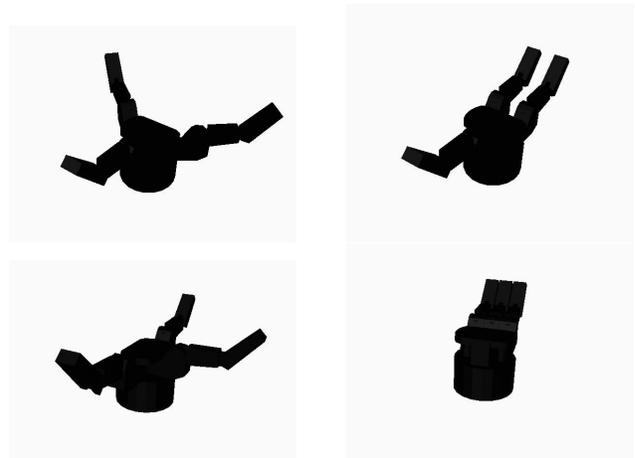

Figure 2: Grasp preshapes for the Barrett hand: spherical, cylindrical, precision-tip, and hook grasps.

2   GraspIt! Overview

GraspIt! is an interactive simulation, planning, analysis, and visualization system for robotic grasping. It can import a wide variety of different hand and robot designs, and a world populated with objects, all of which can be manipulated with in a virtual 3D workspace. A custom collision detection and contact determination system prevents bodies from passing through each other and can find and mark contact locations. The grasp analysis system can evaluate grasps formed with the hand using a variety of different quality measures, and the results of this analysis can be visualized by showing the weak point of a grasp or presenting projections of the 6D grasp wrench space. A dynamics engine can compute contact and friction forces over time, and allows for the evaluation of user written robot control algorithms. Given the system's ability to quickly locate contacts and evaluate grasps, the combination grasp planner/evaluator was a natural extension.

3   Grasp Preshapes

The possible grasp preshapes depends on the complexity of the hand. Our service robot is outfitted with the relatively simple Barrett hand which has only 4 degrees of freedom. It is an eight-axis, three-fingered mechanical hand with each finger having two joints. One finger (often called the thumb) is stationary and the other two can spread synchronously up to 180 degrees about the palm. Although there are eight axes, the hand is controlled by four motors. Each of the three fingers has one actuated proximal link, and a coupled distal link that moves at a fixed rate with the
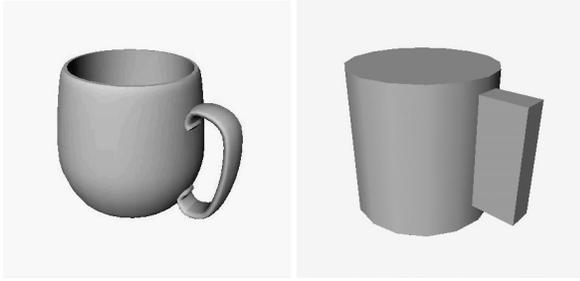
Figure 3: A mug model and its primitive representation. Because most mugs have a similar size and shape, this simplified model can be used for other mugs as well.

proximal link. A novel clutch mechanism allows the distal link to continue to move if the proximal link's motion is obstructed (referred to as *breakaway*). An additional motor controls the synchronous spread of the two fingers about the palm.

For this hand, we have identified four distinct preshapes (shown in figure 2), but only the first two, the spherical and cylindrical configurations, are appropriate for the stable power grasps used in pick and place tasks. A spherical grasp is useful for picking up round objects such as spheres and the top of a cylinder, and a cylindrical grasp, is useful for wrapping around the side of a cylinder or grasping two parallel opposite sides of a box. The precision-tip grasp is best suited for grasping small objects where direct opposition of the fingers is necessary, and the hook grasp may be used to pull a handle or in certain situations as a alternate wrapping grasp when the opposing thumb in the cylindrical grasp would otherwise be obstructed.

4 Grasp Generation

The first step of the grasp planning processes is to generate a set grasp starting positions. To do this, the system requires a simplified version of the object's geometry that consists only of shape primitives such as spheres, cylinders, cones and boxes. The simplified model does not need to match the true object exactly, but the choice of primitives will determine the different strategies used to grasp the object. As an example, we have modeled a coffee mug with a cylinder and a box which roughly approximate the shape and size of the cup and handle (see figure 3).

For each shape, we have defined a set of grasping strategies to limit the huge number of possible grasps. A single grasp starting position consists of a 3D palm position, a 3D orientation which is divided into an approach direction (2D) and a thumb orientation, and a hand preshape.

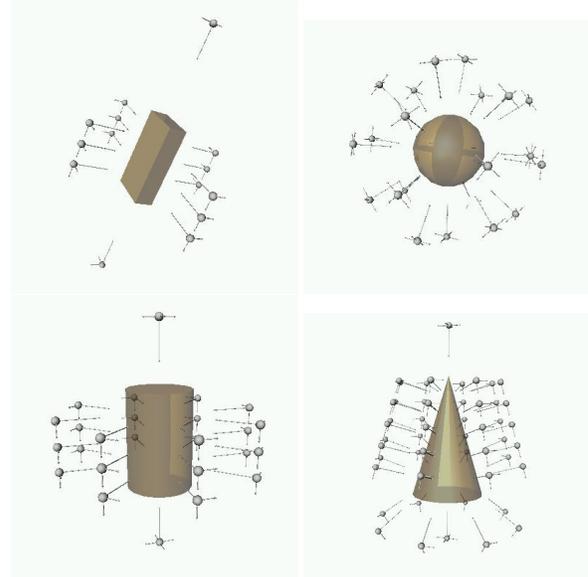**Boxes** should be grasped using the cylinder pregrasp

Figure 4: Examples for grasp generation on single primitives. The balls represent starting positions for the center of the palm. A long arrow shows the grasp approach direction (perpendicular to the palm face), and a short arrow shows the thumb direction (always perpendicular to the approach). In most grasp locations, two or more grasp possibilities are shown, each with a different thumb direction.

shape such that the two fingers and the thumb will contact opposite faces. The palm should be parallel to a face that connects the two opposing faces, and the thumb direction should be perpendicular to the face it will contact.

Spheres should be grasped with the spherical pregrasp shape and the the palm approach vector should pass through the center of the sphere.

Cylinders may be grasped from the side, or from either end.

Side Grasp The cylindrical pregrasp should be used. The grasp approach should be perpendicular to the side surface, and the thumb should either be perpendicular to the central axis of the cylinder, in order to wrap around it, or in the plane containing both the approach direction and the central axis, in order to pinch it at both ends.

End Grasp The spherical pregrasp shape should be used. The palm should be parallel to the end face and aligned with the central axis.

Cones can be grasped in the same ways as a cylinder. However, in the case of a cone with a large radius

and small height, the side grasps will be very similar to a grasp from the top. To handle this, we have added as set of grasps around the bottom rim of the cone, where the palm approach vector is aligned with the bisector of the angle between the bottom face and the side face.

These rules only constrain some of the orientations and positions of the grasp starting locations. We have defined four parameters which control the number of samples chosen in the remaining dimensions:

# of parallel planes For boxes and the side grasps of a cylinder or a cone, this controls how many grasps are planned along the line in the plane of the palm and perpendicular to the thumb. This number is always odd so that a grasp at the midpoint of the face is planned.

# of divisions of 360° For the side grasps of cylinders and cones, this controls how many grasps are planned in a circle lying in each parallel plane. For a sphere, this parameter controls the sampling of both the azimuth and elevation angles.

# of grasp rotations For spheres and the end grasps of cylinders and cones, this controls how many grasps are planned by rotating the palm around an approach vector. This number should not be a multiple of 3 since in the spherical grasp preshape the fingers are separated by 120°, and the grasps would be identical.

# of 180° rotations For boxes and side grasps of cylinders, this number is either one or two, and determines if for each grasp planned, a second grasp should also be planned that is 180° rotation of the cylindrical grasp preshape about the approach vector.

The values of the parameters are automatically chosen based on the dimensions of the object. In the default setting this will lead to 50 to 100 planned grasps for hand sized objects. However, the user can specify that the system should plan fewer or more grasps depending on whether computation time or grasp optimality is more important.

5   Grasp Testing

After the grasp starting positions have been generated, each grasp must be performed and evaluated. Since the grasp evaluation is by far the most time consuming operation, the system checks for infeasible hand configurations at each step of the grasp execution to avoid unnecessary evaluations. In addition, if the hand is connected to a robot arm, any time the arm kinematics prevent the hand from reaching a destination, the grasp is thrown out before evaluation.

To perform a grasp, the hand is first placed at the starting position, and the fingers are positioned in the pregrasp shape. If there are any collisions at this position, the grasp is thrown out and the system proceeds to the next possibility. Next, the hand is moved along the grasp approach direction until it is prevented from moving further by a contact. If the fingers are not blocked by an obstacle, they are closed around the object until contacts or joint limits prevent further motion. If at least one finger is in contact with the object at this point, the grasp is evaluated. If the fingers were blocked from reaching the object by an obstacle, the system backs the whole hand away from the object a small distance and tries the grasp again. This backing off iteration continues until either the fingers reach the object and the grasp can be evaluated or a maximum number of steps is reached.

5.1   Grasp Evaluation

One key feature of this system is that it can be used with any form of grasp evaluation that results in a scalar value. Since our aim is to find stable grasps for pick and place operations, we are using a quality metric that determines the magnitude of the largest worst-case disturbance wrench that can be resisted by a grasp of unit strength. This measure has been proposed in several forms, but it is best described by Ferrari and Canny [4]. The process involves approximating the contact friction cones as a convex sum of a finite number of force vectors around the boundary of the cone, computing the associated object wrench for each force vector, and then finding the convex hull of this set of wrenches. If we assume that each of the contact cones has unit height, then the convex hull corresponds to the L_1 grasp wrench space described by Ferrari and Canny. This space represents the space of wrenches that can be applied by the grasp given that the sum total of the contact normal forces is one. If the origin is not contained within this space, the grasp does not have force-closure (F-C), meaning there exists some set of disturbance wrenches that cannot be resisted by the grasp. In this case the quality of the grasp is 0. Otherwise, the quality of the grasp is equal to the distance from the origin to the closest facet of the convex hull. The wrench in this direction is the most difficult for the grasp to apply.

It is important to note that the amount of friction that can be supported by the contacts greatly affects this quality measure. GraspIt! allows each body to have an associated material type and determines the coefficient of friction for each contact based on a lookup table of material types. In our examples, the links of the Barrett hand are plastic and the objects are either glass or plastic, and the coefficient of

	Tested grasps	Found F-C grasps	Time	Time / F-C grasp
mug	68	44	248 s	5.6 s
phone	52	35	120 s	3.4 s
flask	128	41	478 s	11.6 s
plane	88	19	200 s	10.5 s

Table 1: Performance of the planner with different isolated objects.

	Tested grasps	Found F-C grasps	Time	Time / F-C grasp
mug	68	4	40.4 s	10.1 s
phone	52	1	11.4 s	11.4 s
flask	128	2	232 s	116 s
plane	88	4	49.7 s	12.4 s

Table 2: Performance of the planner with different objects in a complex environment.

friction is either 0.2 or 0.3. If we change the material of the links to rubber (for instance if tactile sensors are mounted on the hand), the coefficient of friction will be 1.0 and the system will find several more force-closure grasps.

6 Planning Results

We have tested the planner with several different objects. The first set of results (shown in figure 5) assumes an object can be grasped from any direction. Note that the model airplane was modeled with only three boxes, which are the dominant features. By not adding boxes for the tail fins, we prevent the system generating and testing grasps of minor elements that will not likely lead to many stable grasps. These tests were all performed on a Pentium IV 1GHz computer, and the planning times for each test are shown in table 1. Next, the hand was attached to the end of a Puma 560 arm model and the objects were placed on a workbench amidst two other obstacles (figures 6 and 7). This reduced the number of feasible grasps and reduced the planning times (table 2).

7 Future Directions

In this paper, we have presented a system that can plan grasps of complex objects given a simplified model of the object built from shape primitives. Using rules defined for these shapes, a set of grasp possibilities, consisting of a
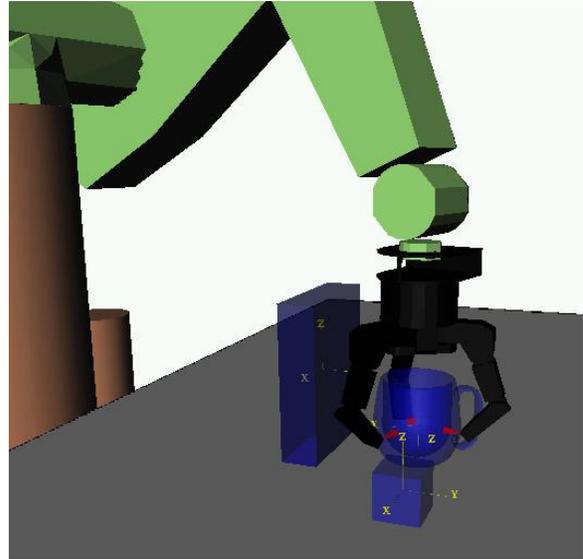

Figure 6: The best planned grasp of the mug in the presence of obstacles and using the reachability constraints of the Puma arm.
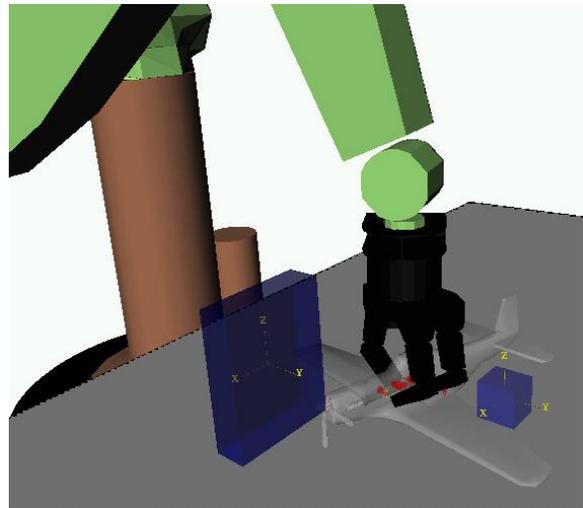

Figure 7: The best planned grasp of the model airplane in a similarly constrained environment.

Figure 5: Planned grasps for a coffee mug, a phone handset, an Erlenmeyer flask, and a model airplane. The first image in each row shows the primitive model used for each object and the generated set of grasps to be tested. The other images show five of the best grasps found sorted in quality order.

location and a grasp preshape, can be generated for the object. Then using the GraspIt! system, these grasps can be tested on the actual object model. This can be done in an isolated setting or with the hand attached to an arm and in the presence of other obstacles.

While this system is ready to be integrated into the planning components of our service robot, there are a few areas that warrant further examination. It would be useful to implement a complete reach planner, so that after a grasp has been planned in a complex environment, we can attempt to find a path back to the robot's current position. Another problem is that for small objects situated between obstacles, it is possible that when the hand closes from the completely open posture the fingers will only collide with the obstacles and never reach the object. However, if a pregrasp also accounted for object size, and the hand started with the fingers already partially closed, it might be able to grasp the object. In addition, it would be useful to generalize the pregrasp postures so that the planner could easily be adapted for use with other robot hands. Finally, there is the issue of where do the primitive models come from? For a service robot, it is not unreasonable to assume it has a database of common objects it must grasp, but for use in more unconstrained environments, we are implementing a vision system that can determine the dominant shapes of an object automatically.

References

[1] C. Borst, M. Fischer, and G. Hirzinger. A fast and robust grasp planner for arbitrary 3D objects. In *Proc. of the 1999 IEEE International Conference on Robotics and Automation*, pages 1890–1896, Detroit, MI, May 1999.

[2] M. R. Cutkosky and P. K. Wright. Modeling manufacturing grips and correlation with the design of robotic hands. In *Proc. of the 1986 IEEE International Conference on Robotics and Automation*, pages 1533–1539, San Francisco, CA, 1986.

[3] D. Ding, Y.-H. Liu, and S. Wang. Computing 3-D optimal form-closure grasps. In *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, pages 3573–3578, San Fransisco, CA, April 2000.

[4] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. of the 1992 IEEE Intl. Conf. on Robotics and Automation*, pages 2290–2295, 1992.

[5] R. D. Hester, M. Cetin, C. Kapoor, and D. Tesar. A criteria-based approach to grasp synthesis. In *Proc. of the 1999 IEEE International Conference on Robotics and Automation*, pages 1255–1260, Detroit, MI, May 1999.

[6] T. Iberall. Human prehension and dexterous robot hands. *The International Journal of Robotics Research*, 16(3):285–299, June 1997.

[7] D. Kragić, A. Miller, and P. Allen. Real-time tracking meets online grasp planning. In *Proc. of the 2001 IEEE Intl. Conf. on Robotics and Automation*, pages 2460–2465, 2001.

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

[9] A. T. Miller and P. K. Allen. GraspIt!: A versatile simulator for grasping analysis. In *Proc. of the ASME Dynamic Systems and Control Division*, volume 2, pages 1251–1258, Orlando, FL, 2000.

[10] B. Mirtich and J. Canny. Easily computable optimum grasps in 2-D and 3-D. In *Proc. of the 1994 IEEE International Conference on Robotics and Automation*, pages 739–747, San Diego, CA, May 1994.

[11] J. Napier. The prehensile movements of the human hand. *Journal of Bone and Joint Surgery*, 38B(4):902–913, November 1956.

[12] N. S. Pollard. *Parallel Methods for Synthesizing Whole-Hand Grasps from Generalized Prototypes*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1994.

[13] J. Ponce, S. Sullivan, J.-D. Boissonnat, and J.-P. Merlet. On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects. In *Proc. of the 1993 IEEE International Conference on Robotics and Automation*, pages 821–827, Atlanta, Georgia, May 1993.

[14] S. A. Stansfield. Robotic grasping of unknown objects: A knowledge-based approach. *International Journal of Robotics Research*, 10(4):314–326, August 1991.