[2] Kenneth Y. Goldberg and Matthew T. Mason. Bayesian grasping. In *IEEE Journal on Robotics and Automation*, pages 1264–1269, 1990.

[3] Kenneth Y. Goldberg and Matthew T. Mason. Generating stochastic parts for a programmable parts feeder. In *IEEE Journal on Robotics and Automation*, pages 352–359, 1991.

[4] W. Eric L. Grimson. On the recognition of curved objects. Technical Report 983, Massachusetts Institute of Technology, 1987.

[5] M. Mani and R. D. W. Wilson. A programmable orienting system for flat parts. In *North American Manufacturing Research Institute*, 1985.

[6] M. T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics and Control*, 5(3):53–71, 1986.

[7] Balas K. Natarajan. Some paragdigms for the automated design of parts feeders. *International Journal of Robotics and Control*, 8(6):98–109, December 1986.

[8] M. A. Peshkin and A. C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal on Robotics and Automation*, October 1988.

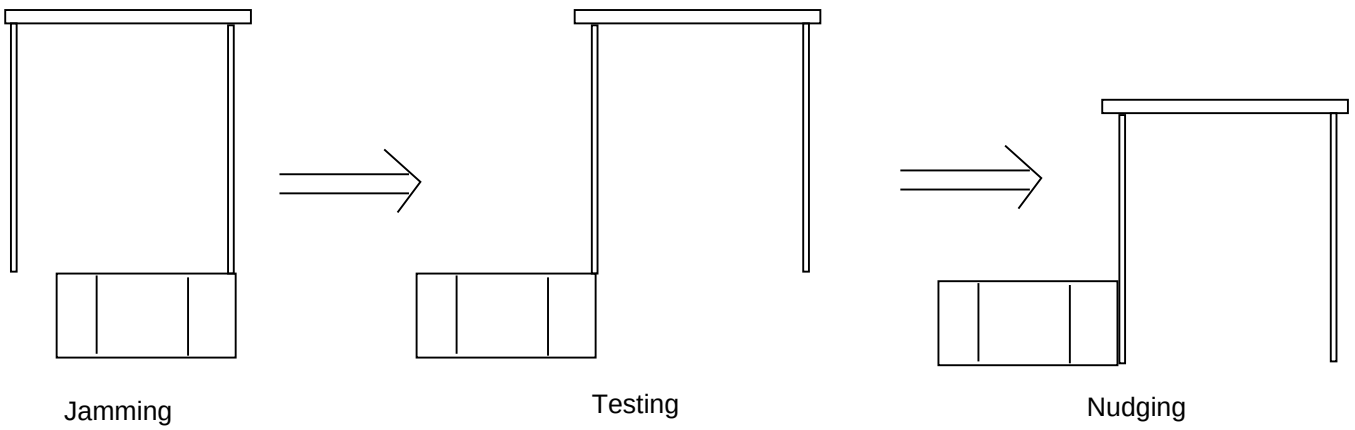

Figure 13: The gripper finger accidentally nudges the object when determing which finger jammed into the object.

The inaccuracy of the sensed diameter made the system extremely unreliable, since this was such a crucial element in the algorithm.

9 Discussion

This algorithm enables the parallel jaw gripper to produce a stable grasp of any object in the presence of position and orientation uncertainty. The algorithm can pick up any object irrespective of its original orientation. The position error, in the most estimate, must be less than half the diameter of the gripper-fingers. This is a conservative estimate because each grasp changes the position of the object. For each grasp instance, the gripper will find the object as long as a part of the object lies under the region between the gripper fingers.

References

[1] M. A. Erdmann and M. Mason. An exploration of sensorless manipulation. In *IEEE Journal on Robotics and Automation*, pages 1569–1574, 1986.

orientation	number of grasps		
(degrees)	Trial 1	Trial 2	Trial 3
0	2	2	2
5	3	2	2
10	2	2	2
15	1	1	1
20	1	1	1
25	1	1	1
30	1	1	1

Figure 11: Results

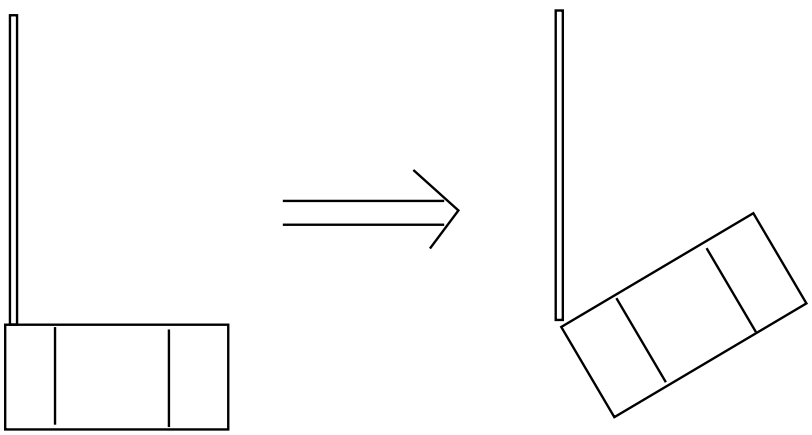

Figure 12: The gripper finger may toss the object (side view)

tolerance correspinds to the accuracy of the sensing. A large number of objects induce a large number of different good distances which leave alot of room for mistaken classification.

4. The most difficult problem that I encountered was transforming the gripper diameter sensor reading into a valid measurement. The error is possibly due to either:

   (a) The gripper fingers were not perfectly flat.

   (b) The sensor sensitivity was approximately 0.002", not relatively small to the objects being grasped.

simultaneously within RobotWorld. The parallel jaw gripper was mounted on the hand which could be lowered and raised. The implementation was written in C, and used a software interface developed by Richard Murray.

The gripper is shown in figure 10.

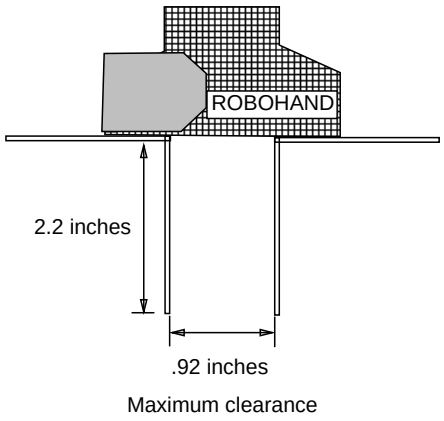

Figure 10: The RobotWorld Gripper

The experiment was to grasp a 3/4" hexnut. In the experiment, the exact position of the hexnut was known, and it was oriented at: $0^o, 5^o, 10^o, 15^o, 20^o, 25^o, and 30^o$.

8 Results

The results of repeated experiments are given in figure 11 below.

1. Sometimes the fingers touched the object so near to its edge that it tossed the object, as illustrated in figure 12.

2. When testing which finger was responsible for jamming into the object, the test finger would nudge the object over, as show in figure 13. This problem persists because of inaccurate knowledge of the distance of the gripper fingers.

3. When the cube (0.55 inches) was jammed between the grippers, the algorithm mistook it for the hexnut (0.75 inches), and exited. The algorithm always assumes that a grasp which matches, with some tolerance, the distance of a good grasp, is the good grasp, where the

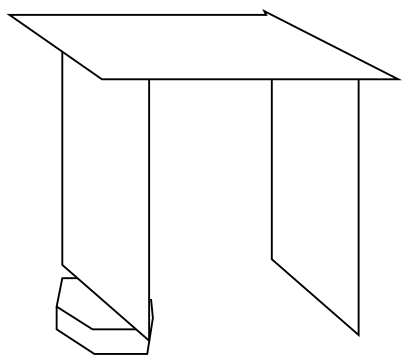

Figure 9: The gripper finger might hit the object

1. The gripper hand is moved up
2. The gripper hand is moved to the right so that the left finger lines up with where the right finger was
3. The gripper hand is moved down
4. If the gripper hand still cannot descend completely, that implies that the right finger was blocked, and therefore the object was lodged under the right finger. (and update the object's position accordingly)
5. Otherwise, if the the gripper hand descends completely, we assume the leftt finger was blocked, and therefore the object was lodged under the left finger. (and update the object's position accordingly)

- The gripper fingers close
- If the grasp is non-optimal, the gripper fingers reopen, and the gripper hand is raised back to a height, GRIPPER_UP

7 Experiment

The algorithm was implemented on RobotWorld and tested on two objects, a hexnut and a cubic die. RobotWorld is a cartesian robotic environment with positional accuracy around 0.001". Multiple modules can be accesed

6 The Algorithm

After a grasp, the diameter of the parallel jaw gripper determines a set of $possible_states$. The algorithm uses a discretized diameter function $\mathcal{SH}_{object,face}(\theta)$ that had been computed off-line. The maximal regrasp angle is found by scanning all possible regrasp angles to find an angle which will separate the $possible_states$. If no single regrasp angle separates all of the possibilities, then pairs of regrasp angles are tried.

6.1 Determining Maximal Regrasp

The sensed diameter of an object, determines a set of possible positions and orientations. This set, called $\mathcal{P}$ is computed by filtering all possible $\mathcal{M}_{object,face}$ shadows, and θ orientations. Define p_i to be a $possible_state$: $(object, face, \theta)$.

- If $\mathcal{P}$ includes a good grasp and non-good grasps, Assume optimistically that one of the good grasps was achieved.
- Otherwise, regrasp at the angle computed off-line which takes into account uncertainty

6.2 Overcoming Position Error

There is a problem that one of the gripper fingers might hit the object, as depicted in figure 9, thus making it impossible to grip the object between the gripper fingers.

When the gripper hand's descent is blocked, it is assumed that the object is blocking one of the gripper fingers. The algorithm program determines which finger the object is blocking, and then recomputes the object's position by adding a small increment; then it tries to grasp the object again.

This part of the algorithm is:

- The gripper hand is lowered down to a height, GRIPPER_DOWN
- If the location of the gripper hand is not GRIPPER_DOWN (with some error tolerance) then:

$$arc1_{radius}\cos(x) = arc2_{radius}\cos(x-(arc2_{center}-arc1_{center}-range_{start}))+\delta$$

$$arc1_{radius}\cos(x) = arc2_{radius}\cos(x-(arc2_{center}-arc1_{center}-range_{start}))-\delta$$

$$arc1_{radius}\cos(x) = arc2_{radius}\cos(x-(arc2_{center}-arc1_{center}-range_{end}))+\delta$$

$$arc1_{radius}\cos(x) = arc2_{radius}\cos(x-(arc2_{center}-arc1_{center}-range_{end}))-\delta$$

3. The extremal values from the previous step define a range of values on $\mathcal{V}_k$, called $newrange$. Intersect $newrange$ with the boundaries of $arc1$. (In figure 8, $newrange$ lies between x and y.)

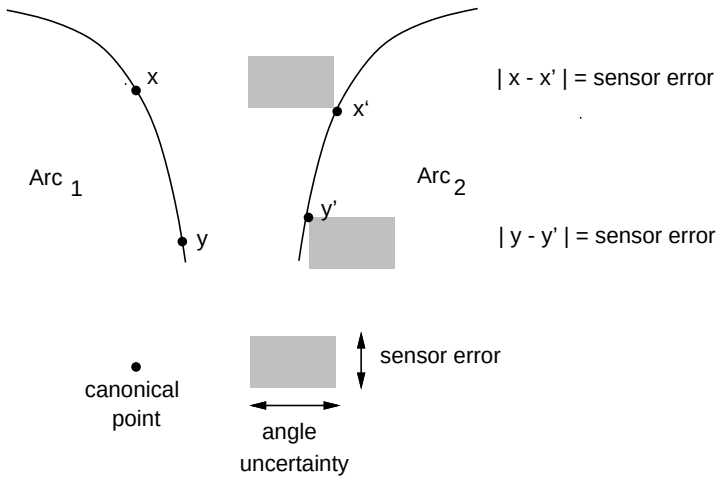

Figure 8: The range of values on arc_1 which are undiscernable from counterparts on arc_2.

4. Recompute the extremal points on $\mathcal{V}_l$ for the endpoints of $newrange$. From these four points and the original ranges: $\mathcal{B}_i, \mathcal{B}_j$, the exact range of translating angles can be computed (since it is continuous). The most distant pair, and closest pair of points define the extremal points of the range of translations that make $\mathcal{B}_i$ and $\mathcal{B}_j$ undiscernible.

5.3 Computing $Confusion[(\mathcal{B}_i \; is \; in \; \mathcal{V}_k), (\mathcal{B}_j \; is \; in \; \mathcal{V}_l)]$

The method for computing bad regrasp angles from the two angle ranges and two semi-arcs is:

1. Compose $\mathcal{B}_i$ amd $\mathcal{B}_j$ to generate an angle range, called *range*. (see Figure 7).

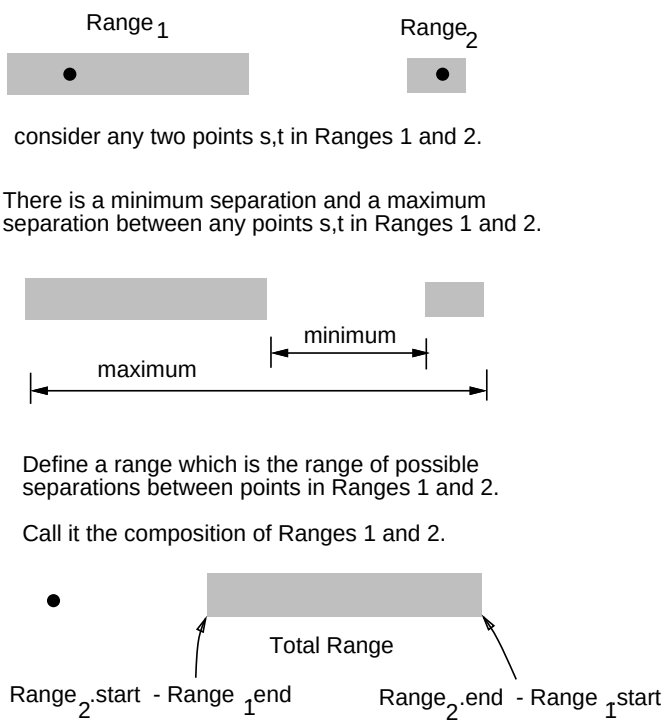

Figure 7: $Range_1 \circ Range_2$, assuming the first angle positioned at the origin, the possible positions for the second angle such that the points are coverable by $Range_1$ and $Range_2$.

2. Using the composite angle range, *range* and the two semi-arcs, $\mathcal{V}_k, \mathcal{V}_l$, compute the **extremal** angles on $\mathcal{V}_k$ for which there exists a counterpart on $\mathcal{V}_l$ that is a valid angular distance away, with difference in heights less than δ. The extremal values are found by solving the transcendental equations, which correspond to the maximal differences, below:

to overcome sensor error. Even with imperfect sensing, **each and every** possible continuous range of re-orientations needs to be discerned. Therefore, the orientation will be exactly known; the gripper will grasp the part stably on the next regrasp.

5.1 Computing Regrasp Angle Into Stable Grasps

To positively end up, there must exist a regrasp angle which translates all of the possible orientations ($\mathcal{B}(object, diameter_{assumed})$) into good grasps ($\mathcal{P}(object)$).

The reorientation angle is computed in the following way:

- For each continuous range of orientations $\mathcal{B}_i$ in $\mathcal{B}$, and each continuous plateau $\mathcal{P}_j$ in $\mathcal{P}$, compute a range of translation angles, $Translate_{\mathcal{B}_i}IntoPlateau_{\mathcal{P}_j}$ which translate $\mathcal{B}_i$ completely within $\mathcal{P}_j$

$$RegraspAngle_{\mathcal{B}_i} = \bigvee_j Translate_{\mathcal{B}_i}IntoPlateau_{\mathcal{P}_j} \tag{3}$$

$$RegraspAngle = \bigwedge RegraspAngle_{\mathcal{B}_i} \tag{4}$$

The center of the largest continuous range of $RegraspAngle$ is used as the regrasp angle to maximize clearance and tolerance.

5.2 Bad Regions of Regrasp Angles

The method of computing the regrasp angle is to compute bad regrasp angles that make two possible ranges, $\mathcal{B}_i, \mathcal{B}_j \in \mathcal{B}(object, diameter)$ undiscernable.

$$BadRegraspAngle = \bigwedge_{i,j \in \mathcal{B}} BadRegrasp(\mathcal{B}_i, \mathcal{B}_j) \tag{5}$$

$$BadRegrasp(\mathcal{B}_i, \mathcal{B}_j) = \bigwedge_{k,l \in \mathcal{V}} Confusion[(\mathcal{B}_i \; is \; in \; \mathcal{V}_k), (\mathcal{B}_j \; is \; in \; \mathcal{V}_l)] \tag{6}$$

All of the arcs $\mathcal{V}$ are split into semi-arcs to simplify the computation. Therefore, each region is continuous and only the extremal values need to be found.

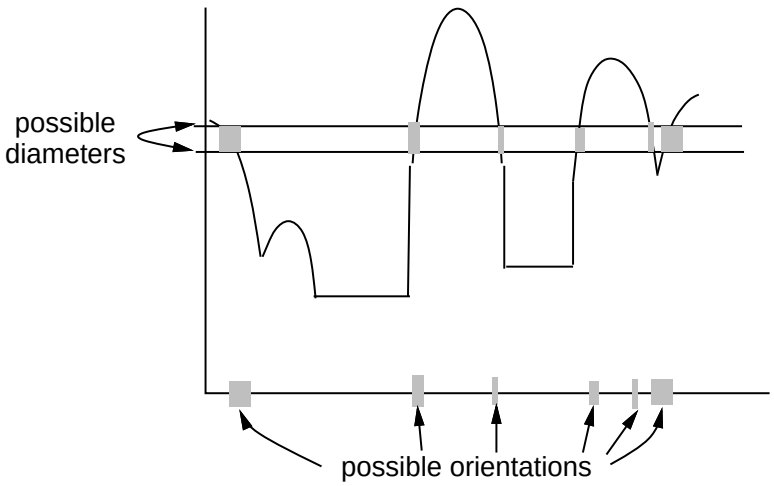

Figure 6: $\mathcal{B}(object, diameter_{assumed})$ derived from a diameter function, and the range of feasible diameters.

- The sensor used was not perfectly repeatable (partially due to the parallel jaw fingers); it may sense either of two possible discrete values. Although the sensor discerned that the distance was within the range $[diameter, diameter + \delta]$, to be conservative, this project assumed that the actual diameter, $diamter_{assumed}$ can lie anywhere inside $\Theta(diameter_{sensed}, \delta)$.

- $\mathcal{B}(object, diameter_{assumed})$, the set of possible orientations consistent with $diameter_{assumed}$, is a disjunction of continuous ranges (see figure 6). $\mathcal{B}$ is pre-defined relative to a specific origin.

- To guarantee correctness, a regrasp angle should guarantee one of two situations"

    1. Translate **every** possible orientation in $\mathcal{B}(object, diameter_{assumed})$ into a stable grasp. $\mathcal{P}(object)$ represents the stable grasps in *object* relative to a pre-defined specific origin. A stable grasp appears like a plateau in $\mathcal{M}$, the modified diameter function.
    2. Translate the possible orientations of the object, $\mathcal{B}(object, diameter_{assumed})$, such that even the closest diameter values will be different enough

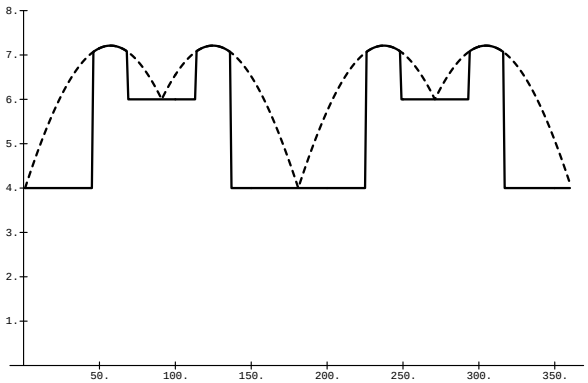

Figure 4: Relationship between Modified Diameter Function (solid) and Actual Diameter Function (dashed) with a 13° friction cone

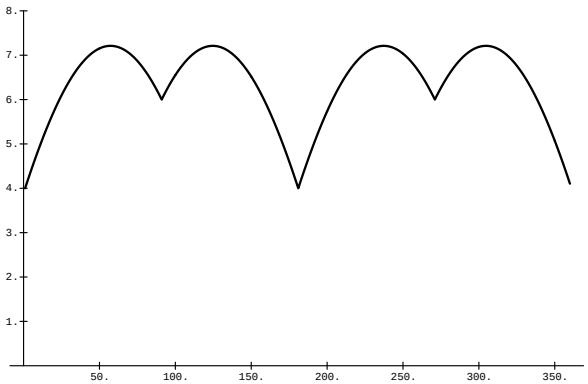

Figure 5: Good, *stable*, Grasps correspond to plateaus of the Modified Distance Function

5 Accounting for Sensor Resolution and Uncertainty

Realistically, the sensors may not be repeatable; returning different values given the same distance: due to resolution (δ), finger bending, and calibration error. The diameters of the possible orientations, after regrasping, should be different enough to be discernible by the inexact sensing device.

- Even if the sensor was repeatable, the resolution would not specify an exact distance, rather a range of possible distances.

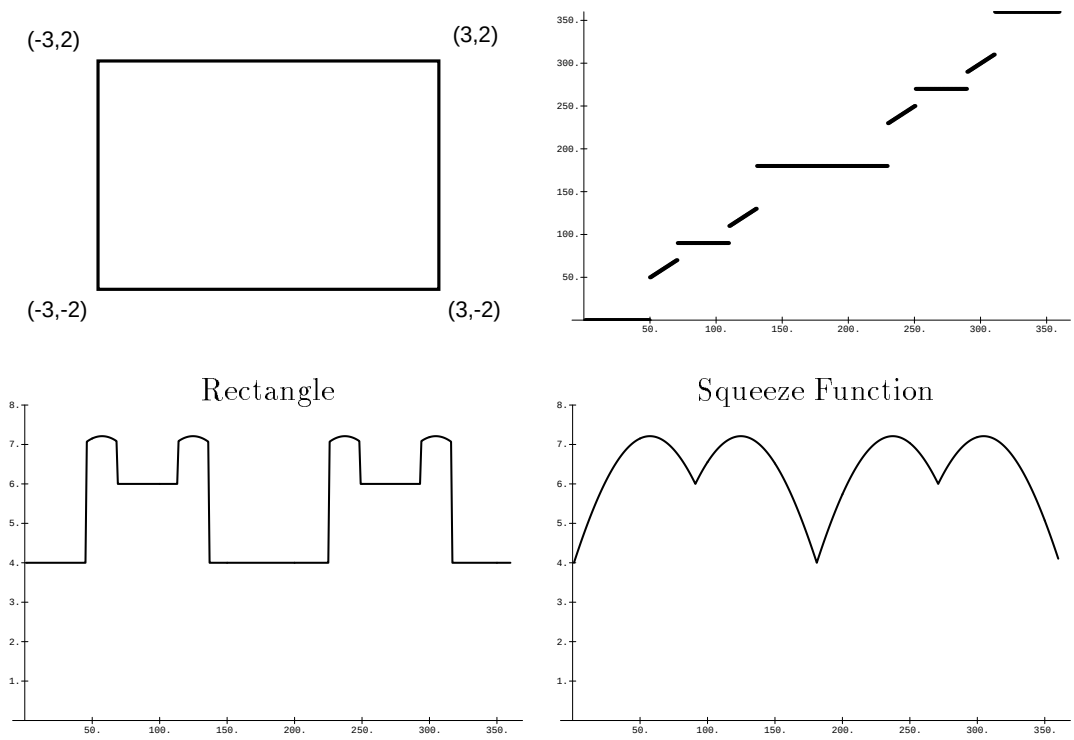

Figure 3: Modified Diameter Function and Diameter Function of a 2:3 Rectangle with a 13° friction cone

The Modified Diameter Function of an object corresponds to the Diameter Function except that vertex-vertex contacts which are not stable by virtue of friction rotate into stable edge contacts, which are the local minima of the Diameter Function.

4.4 Good Grasps

A *good grasp* is a stable grasp, either an edge-edge grasp, or a vertex-vertex grasp which is results from squeezing a number of initial orientations. The good grasps correspond to wide plateaus in the $\mathcal{M}_{object,face}(\theta)$, Modified Diameter Function, and are denoted $\mathcal{P}(object)$ in the next section. Some *good grasps* are shown in Figure 4.

4.3.1 Computing the Modified Diameter Function

To compute $\mathcal{M}$ for a particular planar object, the orientations are categorized by the type of grasp resulting.

1. Edge-edge: Both gripper fingers are in contact with a face of $\mathcal{O}$; i.e. both fingers are in contact with edges of $shadow$, the shadow of $\mathcal{O}$. A stable grasp.

2. Edge-vertex: one finger is in contact with an edge of $shadow$ and the other finger is in contact with a vertex of $shadow$. A stable grasp.

3. Vertex-vertex: Both gripper fingers touch vertices of the object, in contact with vertices of $shadow$. An unstable grasp because the object's orientation is unconstrained.

The Squeeze function applies only when the object is grasped by both fingers. If the object is in an unstable orientation (i.e. the points of contact are not within each other's friction cones) and is being squeezed, the object will be rotated in the direction of less diameter. The object ceases rotating when it reaches a local minimum of diameter.

4.3.2 Equation of Modified Diameter Function

Let $Within_Friction_Cone(\theta)$ be a boolean function that describes whether the parallel gripper contacts of the object at orientation θ are within each other's friction cones.

Let $local_minimum(\theta)$ be a map from θ to the local minimum, θ' found by going in the direction of less diameter.

$$
\begin{aligned}
&Squeeze(\theta) = \\
&If\ Within_Friction_Cone(\theta) \\
&\quad then\ \mathcal{M}(\theta) = \mathcal{D}(\theta) \\
&\quad else\ \mathcal{M}(\theta) = \mathcal{D}(local_minimum(\theta))
\end{aligned}
\tag{2}
$$

The modified diameter function and diameter function of a rectangle of ratio 2:3 is shown in figure 3.

4.3 The Modified Diameter Function $\mathcal{M}_{object,face}$

The $\mathcal{D}$, diameter function for an object is not necessarily the correct tool for simulating parallel jaw grips. As a result of grasping, the object will rotate into a final configuration.

Let $Squeeze_{object,face}(\theta) \rightarrow \theta'$ be a map from an *object* and θ, its initial orientation, to θ' its final orientation.

By composing the $Squeeze$ function with the diameter function, $\mathcal{D}_{object,face}$ we define the modified diameter function, $\mathcal{M}_{object,face}$ of a planar object to be the diameter after being squeezed by the parallel jaw grippers.

$$\mathcal{M}_{object,face}(\theta) = \mathcal{SH}_{object,face}(Squeeze_{object,face}(\theta)) \tag{1}$$

The concept of modified diameter function can be applied to planar objects such as shadows; for planar objects we will drop the subscript and call the modified diameter function $\mathcal{M}$. Similar to the diameter function, computing the modified diameter function of a particular object and face only requires the shadow of the object, *shadow*.

The crucial idea is that the $Squeeze$ function rotates objects which are not stabilized by frictional forces; i.e. contacts which are not within each others friction cones. For these orientations, the object is rotated in the direction of lesser diameter. The $Squeeze$ function only affects contacts which are unstable, as depicted in figure 2.

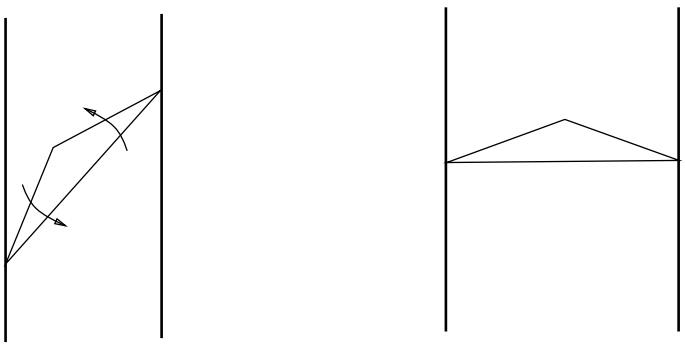

Figure 2: An object cannot be jammed within the Gripper Fingers unless the points of contact are within their friction cones

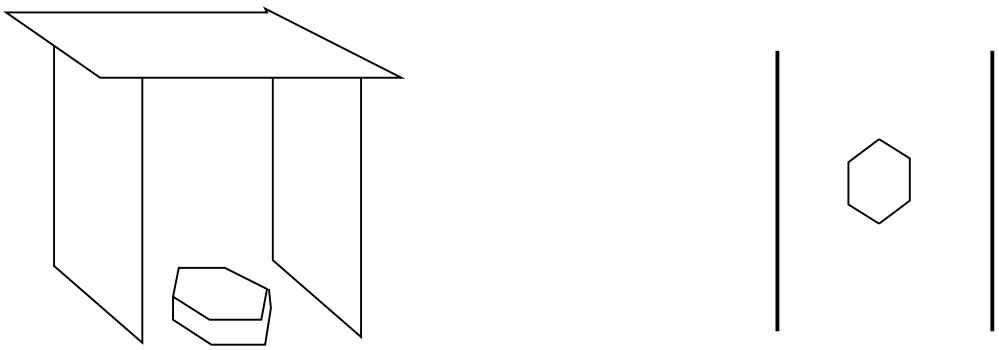

Figure 1: Finding the $\mathcal{D}_\theta$ for a three dimensional object corresponds to finding the $\mathcal{D}_\theta$ for the two dimensional projection, or shadow

2. Since the object is polyhedral, it will always lie with a face of $ConvexHULL(\mathcal{O})$ resting on the floor; this algorithm assumes that the object rests on the same face throughout all of the regrasps.

3. Since the object always rests on a face, there are only a finite number of pairs of angles (ϕ, ρ) that need to be considered. A separate z-axis projection, or shadow $\mathcal{S}$, can be defined for each face.

4. The diameter function $\mathcal{D}_{\theta,\phi,\rho}$ for $\mathcal{O}$ is equivalent to the diameter function $\mathcal{D}, \mathcal{S}_\theta$ for its z-axis projection, or shadow, called $\mathcal{S}$, since it is being grasped by planar gripper fingers [See figure 1]

- $\mathcal{M}_{object,face}(\theta)$ is defined as the modified diameter function of an $object$, the $face$ it is lying on, and the angle θ. The difference between the modified diameter function and the diameter function will be discussed in section 3.3.
- $\mathcal{D}(\theta)$ is defined as the diameter function for a planar $object$.
- $\mathcal{M}(\theta)$ is defined as the modified diameter function for a planar $object$.
- δ is defined as the resolution of the sensor, the discreteness constant.
- $diameter_{sensed}$ is the distance/diameter found by the sensor.
- $diameter_{assumed}$ is the range of possible diameters, by starting with $diameter_{sensed}$ and taking into account sensor inaccuracy.
- $\Theta(x, \delta)$ is the range of values $[x - \delta, x + \delta]$, centered at x and having δ uncertainty.
- $\mathcal{B}(object, diameter_{assumed})$ is defined as the set of orientations that are consistent with possible diameters $diameter_{assumed}$ (relative to a predefined origin). $\mathcal{B}_i$ is the i^{th} continuous region in $\mathcal{B}$
- $\mathcal{P}(object)$ is the set of plateaus, or stable grasp regions of an object (relative to a predefined origin).
- $\mathcal{V}(object)$ is a sinusoidal component in $\mathcal{M}$; used in computing the regrasp angle

4.2 Simplifying Assumptions for Diameter Function $\mathcal{D}_{\theta,\phi}$

The diameter function, $\mathcal{D}(\theta, \phi)$ of a three dimensional object, $\mathcal{O}$, includes extraneous information which will not be necessary in the course of grasping $\mathcal{O}$.

1. Since the parallel jaw gripper has planar fingers, the $ConvexHULL(\mathcal{O})$ determines the diameter function.

3 Simple Sensing Techniques

This project demonstrates the feasability of simple sensing techniques. Complicated methods have been developed which produce the same results, but these involve inordinate amounts of on-line computation. Instead of developing sophisticated algorithms, it is simpler to clear up a small section of the workspace and tailored sensing devices. Simple algorithms are cleaner and faster, and the simple instruments are more robust.

Exploiting the geometric constraints is the best approach to determining the pose of an object. The shadow of an object, resting on a particular face, only has three degrees of freedom, (x, y, θ). Therefore, only three values are sufficient to classify the pose. Three values do not guarantee a simple solution, because of the complex degeneracies within the geometry. I conclude that a relatively few values are necessary, because the rate of degeneracy is not astronomical.

The act of grasping with parallel fingers simplifies the problem of classifying the pose of an object because the fingers can reorient the object into a stable grasp. The algorithm is able to grasp an object irregardless of orientation which initially rests within the circle defined by rotating the fingers.

4 Theoretical Framework

This secion is divided into four parts: describing the notation, stating the assumptions about the diameter function, defining the modified diameter function, and defining good grasps.

4.1 Notation

- $\mathcal{D}_{\theta,\phi}$ $(object)$ is defined as the diameter function of an $object$ and the angles θ and ϕ (where θ is rotation about the z-axis and ϕ is rotation about an axis in the x-y plane).
- $\mathcal{SH}_{object,face}(\theta)$ is defined as the diameter function of an $object$, the $face$ it is lying on, and the angle θ($\mathcal{SH}$ for shadow, as in z-axis projection).
- A $possible_state$ is defined to be the triple $(object, face, \theta)$

1 Introduction

The objective of this project is to stably grasp objects, irrespective of their initial orientation, while overcoming positional and sensing uncertainty. It could also possibly identify an object from a range of possible objects. This project demonstrates the utility of simple sensing devices.

Grasping an object with parallel jaw grippers provides more information than simply that the object is held between the fingers. The distance between the grippers, which is equal to the diameter of the object at its current orientation, constrains the possible orientations. The exact diameter is not known due the sensor's discretization (resolution), as well as sensor inaccuracy; the sensor's result gives a range of possible diameters. When the fingers accidentally land on top of the object, partial information about the location of the object is learned. Computing the optimal regrasp angle by using the sensed diameter of the gripper fingers accounting for sesnor inaccuracy is the major topic of this paper. Sensing when the parallel jaw grippers run into the object is the minor topic.

The previous work section gives background in the field of sensory feedback, and the field of overcoming uncertainty.

2 Previous Work

[6] analyzed planar pushing as foundation for parallel jaw grasping. [5] designed a programmable part orienting system by *pushing* polygonal planar parts. [1] generated a sequence of tilting operations to reduce uncertainty. [7] modelled the orientation as a state, the set of possible orientations as a set of states, and wrote an algorithm to efficiently merged all of the states. [8] designed a programmable part orienting system using a conveyor belt and a series of planar *fences*. [**?**] invented the Squeeze-Grasp Diagram model of parallel jaw grasping. [2] designed a sensorless probabilistic parts orienter by inventing "frictionless" parallel jaw fingers, and also introduced the diameter function into modelling parallel jaw grasps. [3] stochastically generated plans for orienting polygonal parts. [4] used interpretation trees to overcome uncertainty and account for sensor and model error.

Object Localization Using Finger Gap Sensing[1]

Aaron S. Wallack[2] *John F. Canny*[3]

Computer Science Division
University of California
Berkeley, CA 94720

Grasping an object with parallel jaw grippers provides more information than simply that the object is held between the fingers. The distance between the gripper fingers is equal to the diameter of the object at its rotation; thus constraining the possible orientations of the object. By examining the position of the parallel jaw grippers, you can determine whether the gripper fingers ran into the object, i.e. sensing the position of the object. An algorithm was implemented that results in stable grasps of objects irrespective of original orientation, and tolerates some position error, while taking into account sensor inaccuracy and uncertainty.

[1]RAMP 92-2/ESRC 92-3

[2]Supported by Fannie and John Hertz Fellowship,

[3]Supported in part by David and Lucile Packard Fellowship and National Science Foundation Presidential Young Investigator Award (# IRI-8958577).