

Toward Cloud-Based Grasping with Uncertainty in Shape: Estimating Lower Bounds on Achieving Force Closure with Zero-Slip Push Grasps

Ben Kehoe¹ Dmitry Berenson² Ken Goldberg²

Abstract—This paper explores how Cloud Computing can facilitate grasping with shape uncertainty. We consider the most common robot gripper: a pair of thin parallel jaws, and a class of objects that can be modeled as extruded polygons. We model a conservative class of push-grasps that can enhance object alignment. The grasp planning algorithm takes as input an approximate object outline and Gaussian uncertainty around each vertex and center of mass. We define a grasp quality metric based on a lower bound on the probability of achieving force closure. We present a highly-parallelizable algorithm to compute this metric using Monte Carlo sampling. The algorithm uses Coulomb frictional grasp mechanics and a fast geometric test for conservative conditions for force closure. We run the algorithm on a set of sample shapes and compare the grasps with those from a planner that does not model shape uncertainty. We report computation times with single and multi-core computers and sensitivity analysis on algorithm parameters. We also describe physical grasp experiments using the Willow Garage PR2 robot.

I. INTRODUCTION

Although networked robotics has a long history [1], Cloud Computing is a powerful new paradigm for massively parallel computation and real-time sharing of vast data resources. Cloud Robotics has the potential to significantly improve robots working in human environments in four ways: 1) indexing vast libraries of annotated image and object models with information on physical interactions, 2) massively-parallel sample-based motion planning and uncertainty modeling, 3) sharing of outcomes, trajectories, and dynamic control policies for commonly-used robot mechanisms such as Willow Garage’s PR2, 4) obtaining on-demand human guidance when needed.

One area that has not, to our knowledge, been studied is how the second aspect of the Cloud can be used to facilitate grasping objects with uncertainty. Computer vision can approximate geometric shape and pose, and are making great advances in recognition, using indexed databases such as Google Goggles, and these are being integrated with robot manipulation systems [22]. A fundamental challenge, even with perfect recognition, is uncertainty in shape, because of manufacturing tolerances, and mechanics, because of limits on sensing during grasping.

This paper describes initial work toward a method that can leverage Cloud Computing to plan grasps with shape

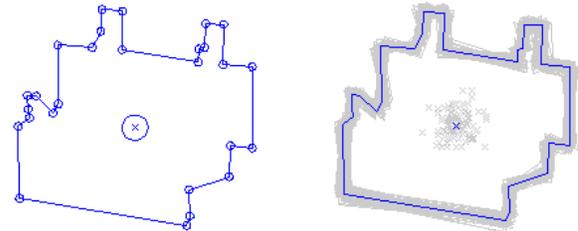


Fig. 1. Shape uncertainty model. On the left, circles with a radius of one standard deviation are drawn around each vertex and the center of mass. On the right, the nominal object is plotted over 100 sampled perturbations.

uncertainty. We currently take a conservative approach: we generate a statistical sample of object shape perturbations to find the value of a quality metric that estimates a lower bound on the probability of force closure for a class of grasps called *zero-slip push grasps*, which can be rapidly evaluated without simulation. We then combine these results, weighted by their probability of occurrence, to estimate a lower bound on the probability of achieving force closure.

We validate our method through extensive testing in simulation and on the PR2 mobile manipulator. Our simulation results suggest that we can generate a set of grasps quickly for a parts with complex geometry. The experiments on the PR2 suggest that our method produces grasps that are significantly more reliable those planned without considering shape uncertainty.

Although our method is not currently implemented in the Cloud, extending it to this domain through parallelization is straightforward, and we demonstrate a proof-of-concept parallel implementation.

II. RELATED WORK

While multi-agent cooperation has a long history in robotics research, only recently has research focused on networked robots sharing information to accomplish tasks widely separated in time and space [26] [38]. The introduction of Cloud Computing can allow computation to be offloaded from robots [4], as well development of databases that allow robots to reuse previous computations in later tasks [11]. Grasping could benefit from this effort, since grasps computed for an object can be applied to similar objects encountered later [10] [16] [18] [27], even for different robots. This allows the construction of grasp databases that can be shared and referenced by multiple robots [18] [22].

Many recent studies in robotic grasping focus on improving grasps on known objects [8] [32] [33] [35] that do not take into account uncertainty. The work in robotic grasping

¹Department of Mechanical Engineering, University of California, Berkeley; Berkeley, CA 94720, USA; benk@berkeley.edu

²Department of Industrial Engineering and Operations Research and Department of Electrical Engineering and Computer Science; University of California, Berkeley; Berkeley, CA 94720, USA; berenson@eecs.berkeley.edu, goldberg@berkeley.edu

that addresses uncertainty largely focuses on uncertainty in part pose [5] [12] [30]. Methods for sensorless part orientation [6] [17] can also be used in the presence of uncertain part pose. However, these methods do not take into account uncertainty in the geometry of the object.

Several studies use contact sensors to improve grasp quality in the presence of uncertain part geometry [15] [19]. However, many robotic grippers do not have contact sensing capability. Even when sensing is not explicitly used to determine geometry, it may be implicitly assumed to be present, as is the case when pinch grasps are required, since the object must not be moved by contact with the gripper [9] [21] [36] [37].

Push manipulation of objects has been extensively investigated by Mason [24] [25] and others [3] [23]. Performing pushing operations with a gripper to reduce pose uncertainty has been demonstrated by Dogar and Srinivasa [12]. However, these methods, again, do not take into account uncertainty in the shape of the object.

One approach to accommodating uncertainty in the part geometry is to use a compliant gripper. Dollar and Howe [13] proposed a four-fingered hand design that uses compliance in the hand to achieve successful grasps without specially considering the geometry of the object. However, the under-actuation of the hand limits its usability with parts that may have fragile regions that cannot be contacted by the gripper.

Saxena et al. [34] used machine learning methods to determine grasp points directly from stereo images of novel objects without an explicit object model, which implicitly assumes object shape uncertainty. Since it does not have an explicit object model, it cannot incorporate prior information about the object such as object shape or uncertainty, either known or estimated. An explicit shape uncertainty model for grasping was proposed by Christopoulos and Schrater [9] approximates the shape boundary with splines but does not account for motion induced by contact from the gripper.

III. PROBLEM STATEMENT

We consider a parallel-jaw gripper, gripping an object from above. We assume that we have a conservative estimate of the coefficient of friction between the gripper and the object, denoted μ .

We assume that the object can be modeled as an extruded polygon to be gripped on its edges, resting on a planar work surface, and that the object has an estimated nominal center of mass, which may not be at the centroid. The gripper-object interaction is assumed to be quasistatic, such that the inertia of the object is negligible [29].

A. Shape Uncertainty Model

Uncertainty in the object shape is modeled as independent, Gaussian distributions on each vertex and center of mass, centered on their nominal values. The variance of the distributions is an input. Other models exist for shape uncertainty [7] [20] that use worst-case bounds rather than probability distributions. One advantage of using probability distributions is that we can use a Monte Carlo approach to evaluate the effect of uncertainty on candidate grasps.

B. Zero-Slip Push Grasps with Force Closure

We consider grasps performed by a parallel-jaw gripper on an object modeled as an extruded polygon. We consider a class of push grasps that enhance object alignment, *zero-slip push grasps with force closure*. We define this as grasps in which the gripper pushes the object without slipping until it rotates into alignment with the first gripper jaw (a *zero-slip push*) and then completes force closure with the second gripper jaw, as seen in Figure 2. Under this conservative definition, any grasp involving slip of either gripper jaw is not included.

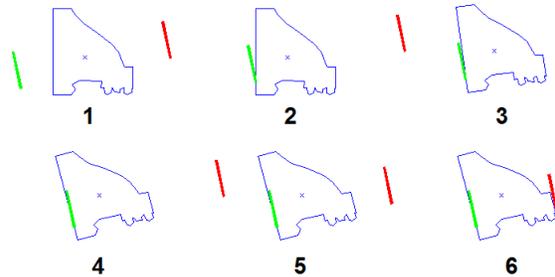


Fig. 2. Snapshots of the execution of a zero-slip push grasp. The green jaw makes the first contact, and the red jaw completes force closure.

The input to the algorithm is a list of edges defining a non-intersecting polygon, denoted S_I , and the variance of the Gaussian uncertainty distributions for the vertices and center of mass.

C. Quality Measure

We define a quality measure $Q(g, S, \theta)$ as a lower bound on the probability that grasp g on object S will result in force closure based on parameter vector θ . The output of the algorithm is $\mathcal{Q} = \{Q(g, S, \theta) \mid g \in G(S)\}$, where $G(S)$ is the set of candidate grasps for object S . The best grasp and Q-value are:

$$g^* = \arg \max_{g \in G} Q(g, S, \theta)$$

$$Q^*(S, \theta) = Q(g^*, S, \theta)$$

D. Candidate Grasp Configuration Space

The configuration space is defined by a starting position and orientation of the first gripper jaw, and a direction of motion from this position. We assume that orientation of the gripper jaw face is perpendicular to the direction of motion.

We reduce the configuration space from three dimensions to two using *nominal contact points* to eliminate some of the redundancies in grasp configurations. Candidate grasps (denoted g_{ij}) are defined as the ordered pair (\hat{p}_i, ϕ_j) , where \hat{p}_i is a *nominal contact point* and ϕ_j is an *approach angle*. A nominal contact point is the point on the boundary of the nominal object that contact by the first gripper jaw would occur. The set of nominal contact points is denoted \hat{P} . An approach angle is an angle from the normal direction of the object boundary (pointing into the shape) at \hat{p}_i . The *approach line* is the line through \hat{p}_i along ϕ_j .

The actual initial contact point for a candidate grasp g_{ij} on a perturbed object may not be near the nominal contact point \hat{p}_i , as the approach line may intersect the object, or some other part of the first gripper jaw may contact the object first. The actual contact configuration corresponding to candidate grasp g_{ij} on object S is $c_{ij,S} = (p, \psi)$, where p is the position of the gripper and ψ is the direction of the jaw relative to the x -axis. Different candidate grasps may have similar contact configurations; for example, if they approach the same edge of the shape. Configurations are grouped into sets of similar configurations denoted $C_{q,\psi}$, where ψ is the same for all configurations in a set and q is a position that describes the set, e.g., one of the p values in the set. The set of similar configuration sets is denoted \mathcal{C} .

IV. GRASP PLANNING ALGORITHM

Our algorithm, shown in Algorithm 1, first filters data to remove noise from the contouring process. Using the resulting filtered polygon, a Monte Carlo analysis is used: it generates a set of candidate grasps, and creates object perturbations drawn from the distribution. For each object perturbation, the candidate grasps are evaluated to estimate if they result in zero-slip pushes (see Section IV-D.1). The successful zero-slip pushes are grouped into sets of similar configurations (see Section IV-D.2), and conservative conditions for force closure are evaluated. Finally, the overall probability of achieving force closure for each candidate grasp is estimated.

Algorithm 1: Zero-Slip Push Grasp Planning Algorithm.
Highlighted line numbers indicate parallelizable steps.

```

1 Filter  $S_I$  into  $S_0$ ;
2 Filter  $S_0$  into  $S_C$ ;
3 Determine nominal contact points  $\hat{P}$  on  $S_0$  using  $S_C$ ;
4 Create candidate grasp set  $G$  from  $\hat{P}$  and  $\Phi$ ;
5 Create object perturbations  $S_1, S_2, \dots, S_N$  of  $S_0$ ;
6 for Object  $S_k = S_1, S_2, \dots, S_N$  do
7   for Candidate grasp  $g_{ij} \in G$  do
8     Estimate if  $g_{ij}$  results in zero-slip push of  $S_k$ ;
9   end
10  Collect similar zero-slip push configurations  $\mathcal{C}$ ;
11  for Similar configuration set  $C_{q,\psi} \in \mathcal{C}$  do
12    Estimate regions of force closure success on  $S_k$ ;
13    for Contact configuration  $c_{q,\psi} \in C_{q,\psi}$  do
14      Find corresponding  $g_{ij}$  for  $c_{q,\psi}$ ;
15      Predict force closure success  $s_{ijk} \in \{0, 1\}$ 
16      of  $g_{ij}$  for  $S_k$ ;
17    end
18  end
19 end
20 for Candidate grasp  $g_{ij} \in G$  do
21   Compute grasp quality  $Q(g_{ij}, S_0, \theta)$ ;
22 end

```

The algorithm uses several parameters, denoted as the vector $\theta = [d_I, d_C, \rho, \Phi, N]$. The shape from the contoured

image, S_I , is filtered to produce a simpler polygon S_0 , and this process takes a parameter d_I that controls the degree of filtering. The uncertainty in object shape and center of mass described in Section III-A are also parameters. Three parameters are used for generation of candidate grasps. A second filtering parameter d_C and a configuration density parameter ρ are used to determine the set of candidate grasp positions, and the set of candidate grasp orientations is a third parameter, denoted Φ . The final parameter is the number of object perturbations, N . We describe these parameters and each step of our algorithm below.

A. Shape Filtering

We assume the input polygon (S_I) is produced from an image of the object using an image contouring algorithm. Filtering is performed on the noisy polygon using an extension of the Ramer–Douglas–Peucker (RDP) algorithm [14] [31]. The RDP algorithm smooths a polyline using a distance parameter (for this step, d_I) that defines the maximum distance a removed vertex can be from the resulting new edge. In our extension to polygons, every pair of adjacent vertices are tested by removing the edge between the vertices, smoothing the resulting polyline, and forming a new polygon with fewer edges by reconnecting the two vertices. The filtered polygon with the fewest edges is selected and becomes the nominal polygon (S_0) for the remainder of the algorithm.

B. Generating Candidate Grasps

After filtering the input polygon, the algorithm then generates a candidate grasp set $G = \{g_{ij} = (\hat{p}_i, \phi_j) \mid \hat{p}_i \in \hat{P}, \phi_j \in \Phi\}$. While each (\hat{p}, ϕ) pair could be independently generated, we use a fixed set of ϕ values as a parameter, and apply them to a generated set of \hat{p} values.

We use a scale-invariant parameter to determine the number of \hat{p} values (i.e., $|\hat{P}|$) for the object, *sample density*, denoted ρ . For each edge, a set of \hat{p} values is generated, linearly spaced with the number of points equal to $\rho \times \frac{\text{length of edge}}{\text{mean edge length}}$. To reduce the effect of complexity on ρ , this is computed on a second filtered shape S_C (using filtering parameter d_C), and the resulting points \hat{P} are projected back onto the nominal shape S_0 . Candidate grasps that are occluded by other parts of the shape are removed.

C. Sampling Object Perturbations

Before testing the candidate grasps, object perturbations are created by sampling from the distributions of each vertex and the center of mass. The number of object perturbations is a parameter to the algorithm; see Section V-D.

D. Evaluating Object Perturbations

For each object perturbation, the candidate grasps are evaluated to estimate whether they achieve zero-slip push grasps with force closure.

1) *Zero-Slip Push Conditions*: The algorithm uses geometric properties of the object to determine all candidate grasps resulting in zero-slip pushes aligned with an object edge for a given gripper width. The conditions for success are as follows: the object purely rotates about the contact point without slipping, the object rotates towards stability with the gripper jaw (that is, the edge becomes aligned with the gripper), and the gripper is not obstructed in becoming aligned with the edge.

As shown by Mason [25], the motion of an object pushed at a given contact point is determined by the friction cone and the direction of pushing. The resulting constraint on candidate grasps is shown in Figure 3.

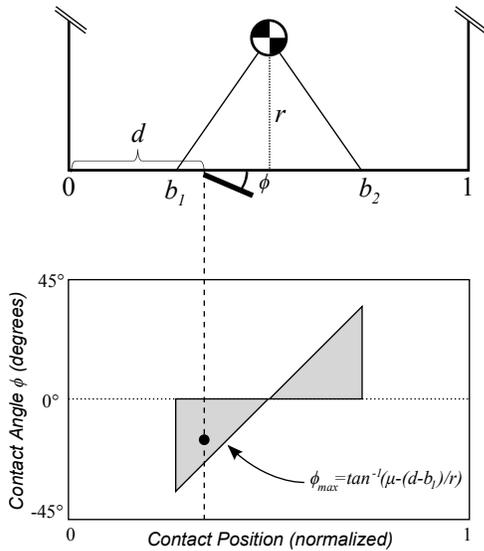


Fig. 3. Configuration space for fast analysis. The upper half of the figure shows a gripper contacting the object at position d with contact angle $\phi = -30^\circ$, with inverse friction cone bounds b_1 and b_2 and perpendicular distance r from the center of mass. Contact with this edge of the object results in the configuration space shown below it; the shaded area is the region where a zero-slip push occurs.

2) *Collecting Similar Zero-Slip Push Configurations*: Before evaluating force closure on the candidate grasps that result in zero-slip pushes, the zero-slip push configurations for those candidate grasps are collected into sets of similar configurations. A similar configuration set often contains all the zero-slip pushes for some edge of the object. Because our estimation of force closure for all positions on an edge can be determined analytically, the estimated closure success of all elements of a similar configuration set can be evaluated simultaneously, as shown in Section IV-D.3.

3) *Conditions for Force Closure*: Force closure on an object is achieved when the line between the contact points on each side lies inside the friction cones of both contact points [28]. If there are multiple contact points on a side, there need be only one successful contact point for successful force closure.

In our algorithm, force closure is considered to be achieved under either of two conditions. First, if the second gripper jaw contacts an edge and the contact direction is within the friction cone, the gripper completes force closure. Second,

if the second gripper jaw contacts a convex vertex, and this convex vertex is opposite a section of the first gripper jaw that contacts the object, force closure is successful.

E. Probability of Achieving Force Closure

Once the candidate grasp conditions have been evaluated for all object perturbations, the probability of achieving force closure for that candidate grasp is estimated using a weighted percentage, where the estimated success or failure on an object perturbation is weighted by the probability of that perturbation occurring.

V. GRASP PLANNING EXPERIMENTS

To test the algorithm in simulation, a set of images of brackets were found on Google Image Search, and manually contoured by tracing a polygon over the image. The shapes produced by this method are shown as objects A through I in Figure 4, along with three simpler, manually-created objects. A comparison with the approach of ignoring uncertainty is presented in Section V-B. We evaluated a large number of parameter combinations, which is detailed in Section V-C. In Section V-E, we report results from testing the parallel aspects of the algorithm.

Except for where noted, tests used vertex variance of 0.2 times the maximum shape radius (measured from the centroid to the vertices), a center of mass variance of 0.7 times the maximum shape radius, a gripper width 25% of the maximum shape diameter (measured between vertices), and a coefficient of friction of 0.7. The tests were run on a Lenovo T420s laptop with a 2.70 GHz processor and 8 GB of RAM, using MATLAB R2011a.

A. Results

For one parameter combination, the full results for two objects are shown in Figures 5 and 6, and best grasp for each of the shapes are shown in Figure 4. The parameters for these figures were $d_C = 0$, $\rho = 1.5$, and $|\Phi| = 5$.

We observed that the algorithm usually did not choose edges close to the center of mass. While this result can seem counterintuitive, grasps close to the center of mass are less robust under our assumptions because an edge close to the center of mass has a smaller region in which zero-slip pushes can be achieved. A perturbation in the center of mass will move this region, invalidating a large number of grasps originally in the region. This effect is seen on Objects C and D. The maximum Q -value on the two horizontal edges of Object D is 43.6. The best grasp on Object J is on an edge close to the center of mass because the edges on either end of the object are too angled to each other for reliable force closure.

Object B, shown in Figure 6, demonstrates the effect of requiring a zero-slip push. Grasps on the edges marked α and β only have very low Q values, because most of each edge is outside the inverse friction cone from the center of mass, meaning any contact will result in slip. The large angle between edges γ and α causes zero-slip pushes on edge γ to fail to achieve our conservative force closure conditions.

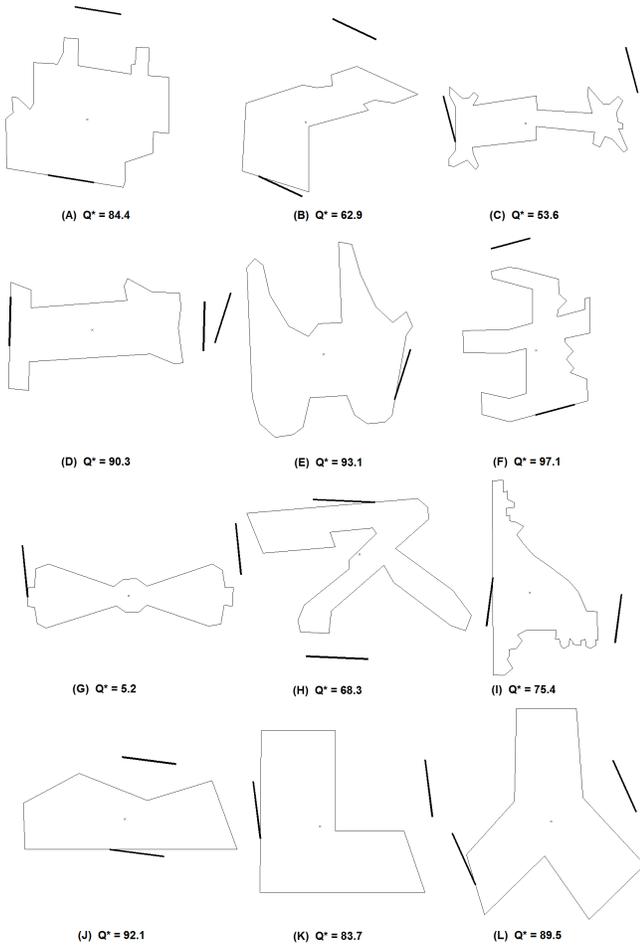


Fig. 4. The test set of brackets. The g^* grasps for parameters $d_C = 0$, $\rho = 1.5$, and $|\Phi| = 5$ are depicted.

However, force closure can be achieved against the vertex labeled τ , and this is reflected in the moderate Q value of some grasps on edge γ .

Object C shows how uncertainty plays a large role in objects with few zero-slip pushes. The grasp shown in Figure 4 closes against protrusions on the opposite end of the object. However, while it has the highest Q value, it is not successful on the nominal shape. The second best grasp contacts the narrow part of the object to the left of the center of mass, and has a Q value of 39.6.

Objects F and H show how the differences in the shape can have a large effect on the quality of grasps, given equal uncertainty. Object F has a very high quality grasp that contacts a flat edge and closes against a small edge with a convex corner. The best grasp on Object H has the same properties, but a much lower Q value. The difference between the objects is that the first edge contacted by the gripper is further from the center of mass on Object F, which as mentioned above can be problematic, and that the uncertainty in the opposite edges on Object H can cause the gripper to contact edges that are more angled.

Object G has a problematic shape for the algorithm. The size of the gripper prevents it from contacting the edges very near the center of mass. The long, straight edges are outside

the inverse friction cone of the center of mass, meaning an contact on them will slip. The ends of the object are narrow and consist of several different edges, which, under perturbation, can prevent zero-slip pushes or force closure from being achieved.

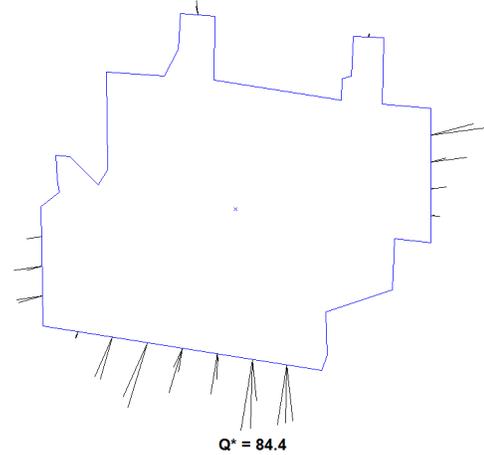


Fig. 5. The algorithm results for object A, using $d_C = 0$, $\rho = 1.5$, and $|\Phi| = 5$. The line segments indicate approach lines for the tested nominal contact points, with the length indicating the Q -value relative to other segments. The approach line with the highest Q -value is labeled.

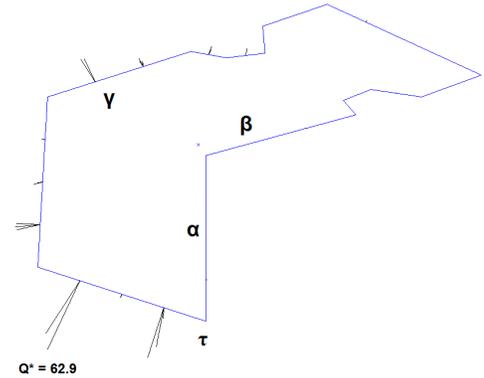


Fig. 6. The algorithm results for object B, using $d_C = 0$, $\rho = 1.5$, and $|\Phi| = 5$. The labels are used in Section V-A to illustrate various aspects of the results.

B. Comparison with Ignoring Shape Uncertainty

We compared our results to a first-order grasp planner ignoring shape uncertainty, which ran the algorithm simply on the nominal object, without considering perturbations. Generally, many candidate grasps are predicted to achieve force closure on the nominal object. However, when subject to uncertainty, many of these grasps become considerably less desirable. For comparison, we ran 84 tests using various parameter values (described in Section V-C), and for each run, the candidate grasps predicted to achieve force closure on the nominal object were tracked and their final quality compared. On average, only 4% of these candidate grasps were also the best grasps after 100 iterations of the algorithm. After 100 iterations, the average Q -value of these candidate grasps was only 58% of the value of Q^* .

Object	Q^*	time (s)	P2	P4	d_C	ρ	$ \Phi $
A	80.8	24.6	1.6	3.0	0.06	1.5	5
A	83.2	31.5	1.8	3.2	0.03	1.5	5
A	85.8	45.9	1.8	3.4	0.09	3	5
A	88.8	242.2	1.8	3.7	0	7.5	9
B	29.0	15.1	1.7	3.0	0.06	1.5	5
B	62.9	21.2	1.7	3.1	0	1.5	5
B	63.0	269.1	1.6	3.7	0.03	20	9
B	67.1	65.7	1.8	3.4	0.03	7.5	5
C	28.6	16.4	1.8	3.4	0.06	1.5	5
C	41.9	24.1	1.8	3.3	0.03	1.5	5
C	59.6	31.9	1.8	3.4	0.09	3	5
C	64.9	142.9	1.9	3.7	0.03	10	5
C	73.5	118.2	1.9	3.7	0.09	7.5	9
C	76.3	466.0	1.8	3.7	0.03	20	9
D	93.1	17.2	1.8	3.1	0.06	1.5	5
D	95.5	37.2	1.8	3.3	0.09	5	5
D	96.3	229.0	1.7	3.6	0.03	20	9
E	92.4	25.4	1.8	3.3	0.06	1.5	5
E	95.2	119.5	1.8	3.5	0.09	15	5
E	96.3	147.6	1.8	3.6	0.03	10	5
F	97.2	31.3	1.8	3.3	0.06	1.5	5
F	97.2	576.1	1.6	3.7	0.03	20	9
F	99.0	134.0	1.9	3.6	0	7.5	5
G	2.6	8.2	1.7	2.8	0.06	1.5	5
G	5.2	25.8	1.7	3.3	0.03	1.5	5
G	13.1	156.4	1.7	3.6	0	7.5	9
G	15.8	103.7	1.7	3.5	0.03	10	5
H	29.4	16.2	1.7	3.2	0.06	1.5	5
H	74.6	23.5	1.7	3.1	0.03	1.5	5
H	74.6	36.0	1.8	3.3	0.09	5	5
H	84.8	91.2	1.8	3.6	0.09	7.5	9
H	87.8	145.5	1.8	3.7	0	7.5	9
I	67.8	50.0	1.9	3.7	0.03	1.5	5
I	75.4	27.0	1.8	3.4	0.06	1.5	5
I	79.3	358.8	1.8	3.7	0	10	5
I	80.2	63.0	1.8	3.6	0.09	7.5	5
I	85.2	792.4	1.4	3.8	0.03	20	9
J	72.7	11.6	1.6	2.8	0.09	3	5
J	92.1	7.7	1.5	2.5	0.06	1.5	5
J	96.0	39.1	1.7	3.2	0.09	15	5
J	97.9	93.3	1.6	3.4	0.03	20	9
J	98.1	45.5	1.7	3.4	0.09	7.5	9
K	12.6	3.8	1.5	2.4	0.06	1.5	5
K	78.1	9.9	1.6	2.8	0.09	3	5
K	98.9	64.1	1.6	3.4	0.06	20	9
L	83.4	16.8	1.7	3.0	0.09	3	5
L	86.5	31.8	1.7	3.1	0.09	7.5	5
L	89.5	8.8	1.5	2.6	0.06	1.5	5

TABLE I

GRASPS PLANNED FOR OBJECTS IN FIGURE 4, SHOWING OBJECT NAME, VALUE OF Q^* , RUNTIME USING MATLAB R2011A ON A 2.70 GHZ COMPUTER WITH 8 GB OF RAM, PARALLEL SPEEDUP FOR TWO AND FOUR CORES, FILTERING PARAMETER d_C , SAMPLE DENSITY ρ , AND NUMBER OF APPROACH ANGLES $|\Phi|$, USING $\mu = 0.7$. P2 AND P4 INDICATE THE SPEEDUPS GAINED FROM RUNNING A PARALLEL IMPLEMENTATION ON THE TWO CORE LENOVO MENTIONED ABOVE, AND A FOUR CORE, 3.4 GHZ COMPUTER WITH 16 GB OF RAM RUNNING MATLAB R2010A, RESPECTIVELY.

C. Sensitivity Analysis

We performed a sensitivity analysis on the parameters for the candidate grasp generation step in the algorithm, which are the maximum distance for the filtering step d_C , sample density ρ , and the approach angle set Φ .

The number of nominal contact points is critical to maximizing the value of Q^* grasps. For a given edge in contact with the first gripper jaw, force closure depends on the opposite edges, which define regions where closure is or is not achieved. With increasing object complexity, the regions become smaller and more numerous, and edges must be covered more densely with contact points to ensure that the regions in which force closure is achieved are found.

To evaluate combinations of values for the parameters, the parameter space was gridded and tested. For filtering, the scale-invariant measure used was fraction of maximum object radius. Increasing values were used until it was judged that large features of the test objects were being filtered out. For the approach angles, a wide, dense range of approach angles were tested, 15 linearly spaced directions from -45° to 45° , inclusive, along with a high value of points per mean edge and no filtering. For all objects tested, the maximum magnitude was never above 13° . We subsequently chose $\pm 15^\circ$ as the range bounds. For sample density (ρ), the value was increased until no further gain in Q^* was seen, and this was used as an upper bound.

The parameter grid included four filtering distances, three sets of approach angles, and seven values for points per mean edge. The filtering distances were 0 (i.e., no filtering other than combining collinear edges), 0.03, 0.06, and 0.09. For approach angles, three sets of linearly spaced points between -15° and 15° , inclusive, were used, with 5, 9, and 15 points, respectively. Only odd values were chosen such that 0° would be included. For sample density, the following seven values were used: 1.5, 3, 5, 7.5, 10, 15, and 20.

The results from the gridded parameter space illustrated the trade-off between Q^* and runtime. Additionally, the discontinuous nature of force closure on polygonal objects was apparent: holding other parameters constant, increasing the sample density sometimes decreased Q^* , when a small region of an edge had the highest probability, and was alternately hit or missed by the spacing of the nominal contact points.

D. Number of Object Perturbations

Reducing the set of object perturbations reduces the runtime of the algorithm, but runs the risk of individual samples having a large effect on the result. We investigated the effect of this trade-off by generating 500 object perturbations and running the algorithm on each sequential subset of 1 to 500 perturbations. The value of Q^* over this range can be seen in Figure 7. In the first few iterations, there are some candidate grasps that are predicted to achieve force closure for all object perturbations tested so far, so the maximum probability is at 1. By the point where 100 object perturbations had been processed, the maximum probability was always within 5% of its value at 500 perturbations.

Additionally, g^* stopped changing before 100 perturbations for all but one object (that is, the best grasp was identified early). This suggests a convergence heuristic: once the Q^* stops changing by more than 5% after testing a new object perturbation, perhaps measured over a moving window, the best grasp has likely been found and the algorithm can terminate.

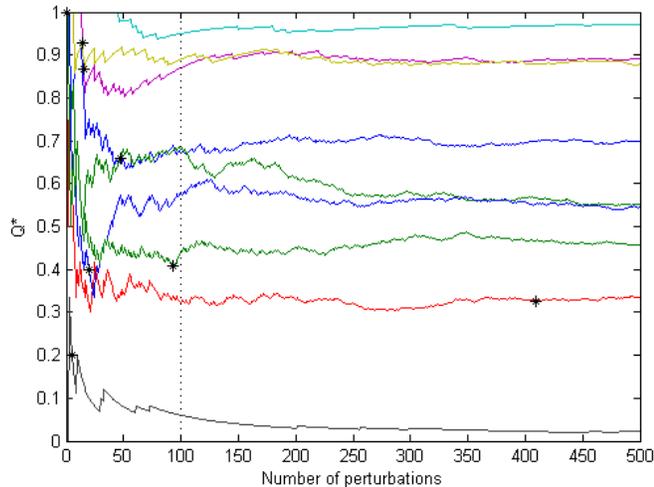


Fig. 7. Q^* vs. number of object perturbations evaluated for objects A-I in the test set. The point at which g^* stops changing is marked with an asterisk.

E. Parallel Tests

Using the MATLAB Parallel Computing Toolbox, the algorithm was tested in MATLAB R2011a on a two core, 2.7 GHz computer with 8 GB of RAM and in MATLAB R2010a on a four core, 3.4 GHz computer with 16 GB of RAM. Parallelization was implemented at the object perturbation step, such that multiple object perturbations could be tested simultaneously. The resulting speedup over the non-parallel running time on the same computer is listed in Table I. The average speedup was 1.7 on the two core computer and 3.3 on the four core. This suggests that, with the limitation of some overhead, the speedup scales with number of cores. Further testing with a massively-parallel implementation in the Cloud could verify this.

VI. GRASP EXECUTION EXPERIMENTS

An object was tested with the Willow Garage PR2 robot [2], a two-armed mobile manipulator. The experimental setup can be seen in Figure 8. Using a whiteboard as a work surface, the object was imaged and contoured to get the shape using the OpenCV image processing library. The algorithm was run using parameters $d_I = 0.002$, $d_C = 0$, $\rho = 10$, and $|\Phi| = 5$.

For Object M, an electrical plug, three representative grasps were tested (shown in Figure 9), with five trial runs each. The first grasp, with $Q = Q^* = 84.4$, achieved force closure for all five trials. The second grasp, with $Q = 54.5$, also achieved force closure for all five trials. The third grasp, with $Q = 23.3$, caused the object to rotate out of alignment



Fig. 8. Experimental setup for Object M.

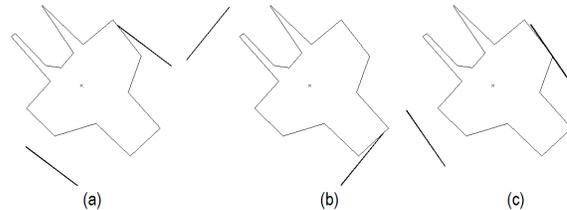


Fig. 9. Grasps tested for Object M.

and failed to achieve force closure for all trials. This grasp failed in the test because of positioning error in the gripper and in the actual center of mass versus in the model.

VII. DISCUSSION AND FUTURE WORK

The algorithm presented in this paper generates samples of object shape perturbations to compute the value of a quality metric that estimates a lower bound on the probability of force closure for zero-slip push grasps. The algorithm is highly parallelizable as described in Section V-E and initial results suggest it could be performed in a Cloud Computing architecture for much greater speedups. A framework like MapReduce may also be used to collect similar stable push configurations.

In future work we will explore how the algorithm might be improved using adaptive sampling, to quickly eliminate poor grasps, and how to relax the zero-slip assumption. We are also interested in seeing how the approach can be used for manufacturing and automation where tolerances are smaller than in the robotics context. The algorithm also may be useful for computing allowable tolerance ranges and for Design for Manufacturability. The results of this paper: code, data, and benchmarks, are available at <http://goldberg.berkeley.edu/archive/cloud-grasping-icra-2012/>.

VIII. ACKNOWLEDGMENTS

We thank Melissa Goldstein, Frank Ong, and Edward Lee, James Kuffner and Frank van der Stappen for valuable discussions, and Pieter Abbeel for allowing us to run experiments on the PR2.

This work was supported in part by NSF Award 0905344.

REFERENCES

- [1] IEEE Society of Robotics and Automation Technical Committee on Networked Robotics. <http://tab.ieee-ras.org/committeefinfo.php?tcid=15>.

- [2] Willow Garage PR2. <http://www.willowgarage.com/pages/pr2/overview>.
- [3] S. Akella and M.T. Mason. Posing polygonal objects in the plane by pushing. In *IEEE International Conference on Robotics and Automation*, pages 2255–2262. IEEE Comput. Soc. Press, 1992.
- [4] Rajesh Arumugam, V.R. Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, F.F. Kong, A.S. Kumar, K.D. Meng, and G.W. Kit. DAVinCi: A cloud computing framework for service robots. In *IEEE International Conference on Robotics and Automation*, pages 3084–3089. IEEE, 2010.
- [5] Dmitry Berenson, Siddhartha S. Srinivasa, and James J. Kuffner. Addressing pose uncertainty in manipulation planning using Task Space Regions. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1419–1425, October 2009.
- [6] R. C. Brost. Automatic Grasp Planning in the Presence of Uncertainty. *The International Journal of Robotics Research*, 7(1):3–17, February 1988.
- [7] Jingliang Chen, Ken Goldberg, Mark H. Overmars, Dan Halperin, Karl F. Böhringer, and Yan Zhuang. Computing tolerance parameters for fixturing and feeding. *Assembly Automation*, 22(2):163–172, 2002.
- [8] Jae-Sook Cheong, Heinrich Kruger, and A Frank van der Stappen. Output-Sensitive Computation of Force-Closure Grasps of a Semi-Algebraic Object. *IEEE Transactions on Automation Science and Engineering*, 8(3):495–505, July 2011.
- [9] V.N. Christopoulos and Paul Schrater. Handling shape and contact location uncertainty in grasping two-dimensional planar objects. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1557–1563. IEEE, 2007.
- [10] Matei Ciocarlie, Kaijen Hsiao, E.G. Jones, Sachin Chitta, R.B. Rusu, and I.A. Sucan. Towards reliable grasping and manipulation in household environments. In *Proceedings of RSS 2010 Workshop on Strategies and Evaluation for Mobile Manipulation in Household Environments*, pages 1–12, New Delhi, India, 2010.
- [11] Matei Ciocarlie, Caroline Pantofaru, Kaijen Hsiao, Gary Bradski, Peter Brook, and Ethan Dreyfuss. A Side of Data With My Robot. *IEEE Robotics & Automation Magazine*, 18(2):44–57, June 2011.
- [12] Mehmet R Dogar and Siddhartha S Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2123–2130. IEEE, October 2010.
- [13] A. Dollar and R. Howe. The SDM Hand: A Highly Adaptive Compliant Grasper for Unstructured Environments. *Experimental Robotics*, pages 3–11, 2009.
- [14] David H Douglas and Thomas K Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, October 1973.
- [15] Javier Felip and Antonio Morales. Robust sensor-based grasp primitive for a three-finger robot hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1811–1816. Ieee, October 2009.
- [16] Jared Glover, Daniela Rus, and Nicholas Roy. Probabilistic Models of Object Geometry for Grasp Planning. In *Robotics: Science and Systems*, Zurich, Switzerland, 2008.
- [17] Ken Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2-4):201–225, October 1993.
- [18] Corey Goldfeder and Peter K. Allen. Data-Driven Grasping. *Autonomous Robots*, 31(1):1–20, April 2011.
- [19] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Grasping POMDPs. In *IEEE International Conference on Robotics and Automation*, pages 4685–4692. Ieee, April 2007.
- [20] Leo Joskowicz, Yaron Ostrovsky-Berman, and Yonatan Myers. Efficient representation and computation of geometric uncertainty: The linear parametric model. *Precision Engineering*, 34(1):2–6, January 2010.
- [21] Ellen Klingbeil, Deepak Rao, Blake Carpenter, Varun Ganapathi, A.Y. Ng, and Oussama Khatib. Grasping with Application to an Autonomous Checkout Robot. In *IEEE International Conference on Robotics and Automation*, 2011.
- [22] James J. Kuffner. Cloud-Enabled Robots. In *IEEE-RAS International Conference on Humanoid Robotics*, Nashville, TN, 2010.
- [23] K.M. Lynch. The mechanics of fine manipulation by pushing. In *IEEE International Conference on Robotics and Automation*, pages 2269–2276. IEEE Comput. Soc. Press, 1992.
- [24] M. T. Mason. Mechanics and Planning of Manipulator Pushing Operations. *The International Journal of Robotics Research*, 5(3):53–71, September 1986.
- [25] M.T. Mason. Manipulator grasping and pushing operations. Technical report, Massachusetts Inst. of Tech., Cambridge (USA). Artificial Intelligence Lab., 1982.
- [26] G. McKee. What is Networked Robotics? *Informatics in Control Automation and Robotics*, 15:35–45, 2008.
- [27] Antonio Morales, Tamim Asfour, Pedram Azad, Steffen Knoop, and Rudiger Dillmann. Integrated Grasp Planning and Visual Object Localization For a Humanoid Robot with Five-Fingered Hands. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5663–5668, October 2006.
- [28] Van-Duc Nguyen. Constructing stable force-closure grasps. In *Proceedings of 1986 ACM Fall Joint Computer Conference*, pages 129–137. IEEE Computer Society Press, 1986.
- [29] M. Peshkin and A. Sanderson. Planning robotic manipulation strategies for sliding objects. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 696–701. Institute of Electrical and Electronics Engineers, 1987.
- [30] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Simultaneous Localization and Grasping as a Belief Space Control Problem. In *International Symposium on Robotics Research*, pages 1–16, 2011.
- [31] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, November 1972.
- [32] Alberto Rodriguez, M.T. Mason, and Steve Ferry. From Caging to Grasping. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2011.
- [33] C. Rosales, L. Ros, J. M. Porta, and R. Suarez. Synthesizing Grasp Configurations with Specified Contact Regions. *The International Journal of Robotics Research*, July 2010.
- [34] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic Grasping of Novel Objects using Vision. *The International Journal of Robotics Research*, 27(2):157–173, February 2008.
- [35] John D Schulman, Ken Goldberg, and Pieter Abbeel. Grasping and Fixturing as Submodular Coverage Problems. In *International Symposium on Robotics Research*, pages 1–12, 2011.
- [36] G. Smith, E. Lee, K. Goldberg, K. Bohringer, and J. Craig. Computing parallel-jaw grips. *IEEE International Conference on Robotics and Automation*, 3:1897–1903, 1999.
- [37] Chao-Ping Tung and A.C. Kak. Fast construction of force-closure grasps. *IEEE Transactions on Robotics and Automation*, 12(4):615–626, June 1996.
- [38] Markus Waibel. RoboEarth: A World Wide Web for Robots. Automaton Blog, IEEE Spectrum. <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/roboearth-a-world-wide-web-for-robots>, Feb. 5, 2011.