Grasp Adjustment on Novel Objects Using Tactile Experience from Similar Local Geometry

Hao Dang and Peter K. Allen

***Abstract*— Due to pose uncertainty, merely executing a planned-to-be stable grasp usually results in an unstable grasp in the physical world. In our previous work [1], we proposed a tactile experience based grasping pipeline which utilizes tactile feedback to adjust hand posture during the grasping task of known objects and improves the performance of robotic grasping under pose uncertainty. In this paper, we extend our work to grasp novel objects by utilizing local geometric similarity. To do this, we select a series of shape primitives to parameterize potential local geometries which novel objects may share in common. We then build a tactile experience database that stores information of stable grasps on these local geometries. Using this tactile experience database, our method is able to guide a grasp adjustment process to grasp novel objects around similar local geometries. Experiments indicate that our approach improves the grasping performance on novel objects with similar local geometries under pose uncertainty.**

I. Introduction

Stable robotic grasping has been one of the most fundamental problems researchers have been working on in the field of robotics. A widely used approach to robotic grasping is to decompose a grasping process into two temporally separated stages: planning and execution. The planning stage is usually done in simulation with the 3D information extracted from a perception system. A stable grasp parameterized by the hand posture and hand-object relative pose is then synthesized. In the execution stage, the planned grasp is sent to a path planner to generate a collision-free trajectory and the robot moves along the newly generated trajectory to the grasping pose. These methods usually use geometrical models of the objects to be grasped in the planning stage. However, since grasp planning is done in a simulation world which is not an exact model of the actual workspace due to imperfect perception and robot calibration, the executed grasps can end up unstable and these methods are sensitive to pose uncertainty.

To cope with pose uncertainty, Berenson et al. used the Task Space Regions (TSR) framework to represent pose uncertainty for planning grasp candidates that are most possible to succeed [2]. Brook et al. analyzed uncertainty in both object identity and object pose for planning the best grasping pose [3]. Stulp et al. designed a framework to generate robust motion primitives by sampling the actual pose of the object from a distribution that represents the state estimation uncertainty [4]. Similarly, Weisz and Allen proposed a new quality metric to measure the robustness of a grasp under object pose uncertainty [5]. Kim et al. considered dynamic movements of the object being manipulated during grasp planning to generate optimal grasp candidates [6].

Both authors are with Department of Computer Science, Columbia University, 450 Computer Science Building, 1214 Amsterdam Avenue, New York, NY, 10027, USA, {dang,allen}@cs.columbia.edu

This work has been funded by NSF Grant IIS-0904514.

Fig. 1. Grasp adjustment using tactile experience, an example that illustrates the progression of hand adjustment. Initially, the grasp (the left column) barely touches one side of the bottle and the finger surface does not align well of the surface of the bottle. After two hand adjustments, the final grasp (the right column) has opposing contacts and the finger surface aligns with the surface of the bottle.

Another group of researchers dealt with uncertainty by considering grasping as a reactive process and designed algorithms to adjust hand posture in an on-line fashion after an initial grasp has been established. Morales et al. used tactile data to cope with uncertainty for the execution of a manipulation task [7]. Platt et al. proposed three variations on null-space grasp control which combine multiple grasp objectives to improve a grasp in unstructured environments [8]. Hsiao et al. used tactile sensing data to estimate hand-object relative pose for synthesizing the next hand trajectory so that a specific grasp can be achieved [9]. Laaksonen et al. proposed a framework to use on-line sensory information to refine object pose and modify the grasp accordingly [10].

In our previous work [1], we developed a grasping pipeline which uses tactile experience for grasping known objects under pose uncertainty. An initial grasp is established using a conventional planning-based grasping pipeline. To ensure the stability of the executed grasp, we developed a grasp adjustment process which analyzes the stability of the grasp and makes necessary hand adjustments. Figure 1 gives one

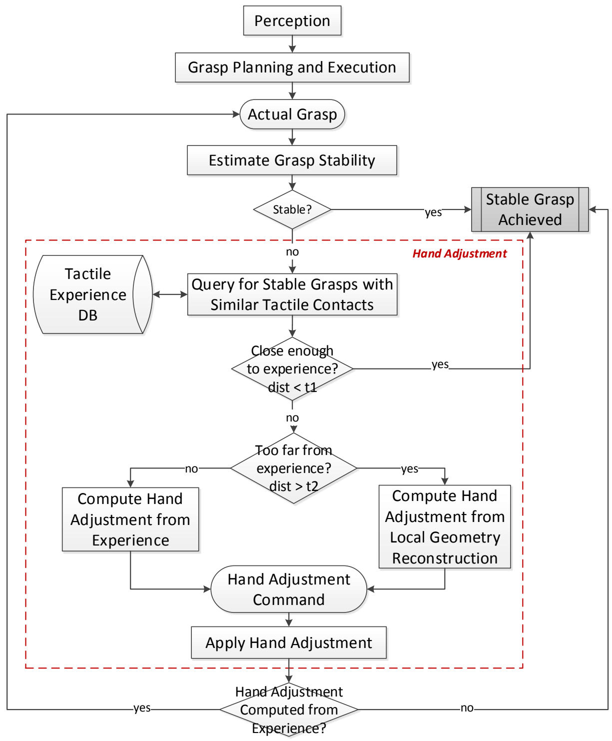


Fig. 2. Our grasping pipeline. A grasping pipeline with a regular planning-based grasp execution procedure and a post-execution grasp adjustment procedure including *Estimate Grasp Stability* and *Hand Adjustment*. A typical planning-based grasping pipeline usually contains only the first two components *Perception* and *Grasp Planning and Execution*. Two thresholds $t_1, t_2, 0 < t_1 < t_2$ were used to evaluate the closeness between two grasps. We will discuss them in Section VI-B.

example of hand adjustments on a Snapple bottle to illustrate how hand adjustments help achieve stable grasps. The hand starts at a grasp where the hand barely touches one side of the Snapple bottle, thus failing to establish opposing contacts. In addition, the palm is not aligned with the vertical direction of the Snapple bottle, resulting in contact surfaces with very limited area. Using a vision system at a distance, this situation is difficult to detect since the pose offset is subtle. However, with tactile sensing, the relative hand pose difference is captured. After two steps of hand adjustments, the grasp is adjusted such that the contacts are opposing each other and the contact surface is increased.

In this paper, we extend our grasp adjustment method to handle novel objects. Our idea is originated from the observation that *objects of different global shapes share similar local geometries*. Thus, tactile experience from similar local geometries can be used for grasping different objects which have local geometries similar to the tactile experience.

II. Our Grasping Pipeline

Figure 2 illustrates the components of our grasping pipeline. Initially, a grasp is applied using modules *Perception* and *Grasp Planning and Execution*, which form a conventional planning-based grasping pipeline. Once the initial grasp is established, the stability of the grasp is estimated by the *Estimate Grasp Stability* module. If the grasp is classified unstable, a hand adjustment will then be synthesized and applied in the *Hand Adjustment* procedure.

In the procedure of *Estimate Grasp Stability*, tactile feedback and hand kinematic information are used to estimate the stability of the grasp [11]. To achieve reasonable hand adjustments, we compute a tactile experience database which consists of a set of stable grasps and use these grasps as a reference to synthesize hand adjustments. The tactile contacts extracted using forward kinematics and tactile sensor readings are used in querying the tactile experience database for stable grasps with similar tactile contacts. If the stable grasps with similar tactile contacts are successfully retrieved, hand adjustment parameters are synthesized and sent to control the hand to make local movements. If there is no similar tactile experience in the database, the local surfaces of the object at contact are reconstructed by moving the hand around to collect tactile contacts on the surface and stable grasps are planned based on the reconstructed local geometry.

A hand adjustment specifies the changes to the current grasp. It consists of changes in hand location, orientation, and the selected degrees of freedom (DOF) to control*. We can write it compactly as $Adj =< p, o, s >$, where $p \in R^3$ is a 3-D vector specifying the new hand position in the current hand coordinate system, $o \in S^3$ is the new hand orientation in the current hand coordinate system represented as a quaternion, and $s \in R^{|S_{dof}|}$ is a vector storing value changes for the set of selected DOFs, S_{dof}, which we want to control.

III. Tactile Experience Database

A tactile experience database consists of stable grasps and their corresponding tactile contacts. It provides precomputed knowledge about the tactile contacts a stable grasp should contain. A grasp, $\mathcal{G}$, in the tactile experience database can be considered as $\mathcal{G} = \{\mathcal{P}, \mathcal{J}, \mathcal{T}, \mathcal{C}, \mathcal{L}\}$ where $\mathcal{P} =< p, o >, p \in R^3, o \in S^3$ specifies the hand pose in the object coordinate system, including the position and orientation of the hand. The orientation is represented using quaternions. $\mathcal{J} = \{j_1, j_2, ..., j_N\}, j_i \in R$ records the N joint angles of the grasp. As an example, for a Barrett hand, we can choose $N = 7$ and record the 7 joint values. $\mathcal{T} = \{t_1, t_2, ..., t_K, t_i \in R\}$ is the K tactile sensor readings. As an example, for a Barrett hand, there are 24 tactile sensors on each fingertip and the palm. Since it has three fingers and one palm, $K = 96$. $\mathcal{C} = \{c_1, c_2, ..., c_M\}, c_i =< p_i, o_i >, p_i \in R^3, o_i \in S^3$ is the set of tactile contacts, indicating the locations, p_i, and the orientations, o_i of the M activated tactile sensors. $\mathcal{L} = \{\mathcal{G}_i^l | \mathcal{G}_i^l = \{Adj, \mathcal{J}, \mathcal{T}, \mathcal{C}\}\}$ is the local tactile experience which stores related information for perturbed grasps within the neighborhood of stable grasp $\mathcal{G}$. Local experience can be used to better locate a grasp within the neighborhood of the corresponding stable grasp based on which the local experience is generated. Adj stores the inverse of the perturbation from the stable grasp to a perturbed grasp. Using this transformation Adj, we can adjust a perturbed grasp to achieve the corresponding stable one.

*Usually, a robot hand contains several DOFs, but we only want to control a subset of these DOFs during a hand adjustment procedure.

IV. Hand Adjustment

We now describe the hand adjustment procedure in Figure 2 and explain the steps to compute a hand adjustment.

A. Querying for Stable Grasps with Similar Tactile Contacts

The first step is to extract the tactile contacts from an actual grasp. This can be done using forward kinematics of the robot hand. Once the set of tactile contacts are obtained, we query the tactile experience database for stable grasps that share similar tactile contacts. To this end, we define a distance function which measures the similarity between two grasps $\mathcal{G}_1$ and $\mathcal{G}_2$. This distance function considers both the tactile contact configuration and the hand posture between two grasps. In our work, we only use the location of a contact in the distance metric. The distance metric can be expressed as

$$dist(\mathcal{G}_1, \mathcal{G}_2) = \alpha||js_1 - js_2|| + \frac{1}{2} \cdot \sum_{m=1}^{N_1} \min_n (||c_m^1 - c_n^2||) + \frac{1}{2} \cdot \sum_{m=1}^{N_2} \min_n (||c_m^2 - c_n^1||) \quad (1)$$

where c_m^i is the m^{th} contact of the grasp i, N_i is the number of contacts of grasp i, and js_i is the joint values for the selected DOFs of the grasp i. α is a scaling factor for the Euclidean distance between the joint angles of the selected DOFs. The first part of the right side of the equation measures the difference between the joint angles for the selected DOFs. The last two parts measure the Euclidean distance between the two sets of contacts in terms of their positions. We also apply this function to measure the distance between a local tactile experience entry $\mathcal{G}_i^l$ and a grasp $\mathcal{G}$ using $dist(\mathcal{G}_i^l, \mathcal{G})$ where the values of $\mathcal{G}_1$ in Equation 1 come from the grasp of $\mathcal{G}_i^l$.

B. Hand Adjustment from Tactile Experience

All the k nearest neighbors are stable grasps and they share similar tactile contacts with the actual grasp. In this case, it is reasonable to assume that the local geometry is similar where the contacts are established. Although the actual grasp shares similar tactile contacts with stable grasps, it is not close enough to be a stable one. However, it is possible that this grasp is away from a stable grasp by a small offset transformation. The goal of this step is to synthesize this offset transformation and generate a hand adjustment to optimize the grasp towards a stable one.

Algorithm 1 outlines the search for a hand adjustment command in Figure 2 using tactile experience. The idea in this algorithm is to use the tactile experience to locate the actual grasp around each of the k nearest neighbors (stable grasps) and synthesize a hand adjustment based on the offset transformations from them. The first step of this algorithm is to look into the tactile experience database and locate the top k stable grasps that share similar tactile feedback (Line 1). Since the actual grasp shares similar tactile feedback as these k stable grasps, the actual grasp is probable to be within a small neighborhood of some of these stable grasps. From

Algorithm 1: Compute a hand adjustment (see [1] for more details)

Input: A robotic grasp $\mathcal{G}_x$, and a tactile experience database

Output: A hand adjustment $Adj =< p, o, s >$

1 Locate k nearest neighbors to $\mathcal{G}_x$, $List^* = \{\mathcal{G}_1, ...\mathcal{G}_k\}$ according to $dist(\mathcal{G}_i, \mathcal{G}_x)$
2 $reference\_dist = [\ ]$, $experience\_database = [\ ]$
3 **foreach** $\mathcal{G}_i \in List^*$ **do**
4   Obtain local tactile experience of $\mathcal{G}_i$, $local\_exp$
5   Rank $local\_exp$ based on $dist(\mathcal{G}_j^l, \mathcal{G}_x)$ where $\mathcal{G}_j^l \in local\_exp$
6     $experience\_database.append(local\_exp)$
7     $reference\_dist.append(\sum_{j=1}^{j \leq 5} dist(\mathcal{G}_j^l, \mathcal{G}_x))$ where $\mathcal{G}_j^l \in local\_exp$
8 **end**
9 $min\_ind = \arg\min_{ind}(reference\_dist[ind])$
10 $experience = experience\_database[min\_ind]$
11 $Adj^* = WeightedTransformation(\mathcal{G}_x, \{\mathcal{G}_j^l | \mathcal{G}_j^l \in experience, 1 \leq j \leq 5\})$
12 Return Adj^*

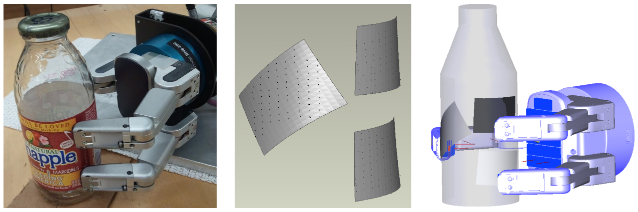
Line 3 to 8, we look into the neighborhood of each of the k stable grasps and try to evaluate how well the actual grasp can be located within the neighborhood of each stable grasp using the distance function as in Equation 1. The refined search within the neighborhood of each stable grasp provides detailed relative information of the actual grasp with respect to the stable grasp. In Line 9 to 10, we decide the stable grasp within whose neighborhood we can best locate the actual grasp. Then the weighted transformations of the perturbed grasps within this neighborhood is calculated in Line 11 and is returned as the hand adjustment. For more details, interested readers please refer to our previous work in [1].

C. Hand Adjustment from Local Geometry Reconstruction

When the actual grasp is far away from any stable grasps in the tactile experience database, there will be no similar tactile experience found in the database. In this situation, a local geometry exploration will take place to reconstruct the local geometry around each of the contacts between the hand and the object. Sample points on the surface of the object are extracted from activated tactile sensors while the hand is moving within the neighborhood of the initial grasping pose. It is assumed that a local geometry is smooth and can be represented using a quadratic function as follows

$$z = \alpha_{20}x^2 + \alpha_{11}xy + \alpha_{02}y^2 + \alpha_{10}x + \alpha_{01}y + \alpha_{00} \quad (2)$$

Fitting the point cloud to the quadratic function above is an optimization process. We use *levmar*, an open source implementation of Levenberg-Marquardt nonlinear least squares algorithms in C/C++, to find the optimal parameters of the function [12]. With a set of optimal parameters, we can approximate the local geometry and synthesize a mesh for



(a) (b) (c)

Fig. 3. Grasping using grasps planned on reconstructed local geometry. Figure 3(a) is a snapshot of executing the grasp planned using the *GraspIt!* simulator [14] on the local geometry shown in Figure 3(b). In Figure 3(b), the reconstructed local geometry is shown in gray with original data points extracted from tactile sensing data shown in black. Figure 3(c) shows the planned grasp, where the transparent bottle model is solely for visualization.

each contact. Once we have reconstructed a mesh model of the local geometries, we use the Eigengrasp planner [13] to search around the local geometry and plan stable grasps on the local geometry. Figure 3 shows an example of a Barrett hand executing a grasp after it has reconstructed the local geometry of a Snapple bottle and planned a stable grasp using the reconstructed local geometry.

D. Apply Hand Adjustment

Once a hand adjustment command $Adj^* =< p, o, s >$ is found, the following hand adjustment is accomplished in three steps. First, the hand opens its fingers so that it lets the object go and backs up to have some safe margin between the palm and the object. Second, the selected DOFs change to the values specified by s. The hand moves to a location 5 centimeters (subject to change for different hands) away from the goal position with the goal orientation o. Third, the hand moves in guarded mode towards the goal position. The hand will either reach the goal position or stop if it contacts anything before it reaches the goal.

V. A Tactile Experience Database for Novel Objects

The tactile experience database stores stable grasps as tactile experience. The tactile experience is used to infer hand-object relative pose and make necessary hand adjustments. At the core of our work, the local geometry at contact is the focus and there is no assumption made about the global shape of an object. Thus, our methods are global-shape-independent. Considering the fact that different objects with different global shapes may share similar local geometries, if we can obtain a tactile experience database built on a set of local geometries that are shared by different objects, our methods can be extended to grasp novel objects on their similar local geometries.

For example, the white paint bottle and the wineglass in Figure 5 have different global shapes but they share similar cylindrical local geometry. In this sense, if we have a stable grasp on such a cylindrical geometry in our tactile experience database, we can use our method to guide a grasp adjustment procedure on objects with such cylindrical local geometries.

In order to utilize the similarity of local geometries across objects with different global shapes, we need to have a reasonably good parameterization of local geometries. To this end, we chose a series of shape primitives and use them to build our tactile experience database. Figure 4 shows examples of the four types of the shape primitives we used to generate our tactile experience database: boxes, cylinders, ellipsoids, and spheres. For each type of shape primitives, we collected different object models with different dimensions as illustrated in the second column. For each of the object models, we also define stable grasps. Examples of stable grasps we defined on these shape primitives are illustrated in the third column of Figure 4. These stable grasps are used in our tactile experience database as stable grasp experience, which guides a grasp adjustment process on similar local geometries. In Section VI-B, we will discuss how we generate our tactile experience database using these shape primitives.

Shape primitive	Model variance	Stable grasps
box	width 30mm ~ 100mm (interval: 5mm)	
cylinder	diameter 30mm ~ 100mm (interval: 5mm)	
ellipsoid	semi-principal axis b 30mm ~ 100mm (interval: 5mm) (a = 100mm)	
sphere	diameter 50mm ~ 100mm (interval: 5mm)	

Fig. 4. Shape primitives used to generate a tactile experience database for grasping novel objects.

VI. Experiments

A. Experimental Setup

In our experiment, a Barrett hand is used as the robot hand. The selected DOFs s in a hand adjustment $Adj =< p, o, s >$ controlled the spread angle of the Barrett hand. We chose five commonly seen objects as our test objects shown in Figure 5: a box, a paint bottle, a wine glass, a mug, and a canteen. We assumed a table-top grasping scenario where the objects rest on a flat surface. In this situation, the pose error can be parameterized by $< x, y, \theta >$, where x and y are the two orthogonal directions defining the table plane and θ represents the rotation around the normal to the table plane. In our experiments, we intentionally generated a list of pose error with an approximately uniform distribution over $x \in [-30, 30]$ in millimeter, $y \in [-30, 30]$ in millimeter, and $\theta \in [-20, 20]$ in degree. By injecting different pose errors into a stable grasping pose, we could perturb the stable grasp from its ideal grasping pose and generate grasping scenarios with different pose uncertainty.

B. Building A Tactile Experience Database

To build our tactile experience database, we first defined a stable grasp for each shape primitive in Figure 4 using

Fig. 5. Novel objects used in the experiments: a box, a paint bottle, a wineglass, a mug, and a canteen.

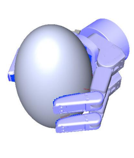
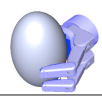
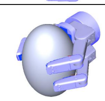
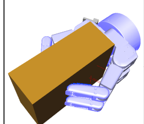
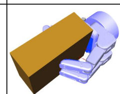
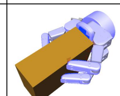
Stable Grasp	Examples of Precomputed Tactile Experience		
	Pose Error	Perturbed Grasp	Tactile Experience (Contact List)
	<15,15,10>		{19.225, 34.5239, 88.0005}, {19.225, 32.186, 92.399}, {-30.775, 42.5214, 89.9098}, {-25.6, 42.5214, 89.9098}, {-20.4, 42.5214, 89.9098}, {-25.6, 40.6573, 94.5291}, {-20.4, 40.6573, 94.5291}, {0.6, -71.7234, 78.6886}, {-4.6, -71.7234, 78.6886}, {0.6, -70.7011, 83.5638}, {-4.6, -70.7011, 83.5638}
	<-20,-10,10>		{19.225, 64.368, 79.3297}, {24.4, 64.368, 79.3297}, {19.225, 62.0501, 84.9535}, {24.4, 62.0501, 84.9535}, {-20.4, 64.1653, 77.7957}, {-20.4, 61.1092, 82.959}, {-20.4, 58.9136, 88.6317}, {5.775, -43.6189, 90.3828}, {5.775, -41.8205, 95.028}, {5.775, -42.1305, 99.4401}
	<15,15,10>		{19.225, 13.1849, 89.2469}, {19.225, 12.1789, 92.1843}, {-30.775, 21.9145, 93.3284}, {-30.775, 21.2446, 96.3603}
	<-20,-10,10>		{19.225, 38.4799, 94.06}, {19.225, 38.6096, 98.481}, {-30.775, 47.2648, 84.1043}, {-25.6, 47.2648, 84.1043}, {-30.775, 46.7495, 87.1661}, {-25.6, 46.7495, 87.1661}, {5.775, -22.5802, 93.8677}, {5.775, -21.9356, 96.905}

Fig. 6. Examples of stable grasps and precomputed local tactile experience. Each stable grasp in the tactile experience database is stored with a complete set of parameters that can be used to reconstruct the grasp, including the joint values and hand pose with respect to the object. The local tactile experience for nearby perturbed grasps is precomputed based on a list of precomputed pose error for the object, which is described in Section VI-A.

the *GraspIt!* simulator [14]. For each of the stable grasps stored in the tactile experience database, we precomputed the tactile feedback at grasping poses perturbed from each of the stable grasps due to pose error. To do this, we first put the hand at the ideal grasping pose. Then, we uniformly sampled the space of pose uncertainty $S = \{< x, y, \theta > | x \in [-30, 30], y \in [-30, 30], \theta \in [-20, 20]\}$ and used each of the sampling pose error $< x, y, \theta >$ to perturb the object and generate the tactile feedback at each of the perturbed grasping pose. In our work, the sampling is 5 millimeters in dimension-x and dimension-y and 5 degrees in dimension-θ. For the spread angle, we sampled 5 degrees above and below the ideal spread angle for the grasp. Thus, this precomputation generated 4572 sampling perturbed grasping poses for each stable grasp. This precomputation took place off-line and the database was stored for later use. Figure 6 gives us two examples of stable grasps on two different objects and four exemplar local tactile experience records generated from the corresponding pose error. These stable grasps along with the tactile feedback from the perturbed grasping poses were stored to form our tactile experience database.

In terms of the parameter in the distance function, Equation 1, we empirically chose the value $\alpha = 100$ so that 0.01 radian difference in joint angles is equivalent to 1 mm in Euclidean distance. We also experimentally chose two thresholds for the decision diamonds in the *Hand Adjustment* procedure of Figure 2. If the distance metric of an actual grasp to one of the nearest neighbors in the database is less than $t_1 = 10.0$, we decide this grasp is close enough to experience. If the distance metric of an actual grasp to any one of the nearest neighbors is greater than $t_2 = 30.0$, we decide the actual grasp is too far from experience and no similar experience is found.

C. Grasping Novel Objects on Similar Local Geometries

The objects we selected as shown in Figure 5 are novel in terms of the fact that their global geometries are different from the objects used to generate our tactile experience database. However, these objects share similar local geometries with the objects in our tactile experience database. For example, the body of the wineglass is similar to both an ellipsoid and a cylinder. Thus, when grasping the similar local geometry, we can expect similar tactile feedback from the tactile sensors. To generate a list of initial grasping poses for experiment, we first predefined a stable grasp on each object around the similar local geometry. We then injected 10 pose errors to each of these stable grasps. Since no models of these novel objects were obtained for pose estimation through vision, we put each object at a predefined known location and hard-coded the pose of each object. It is worth noting that although we manually chose the initial grasping pose based on geometric similarity, this process can be automated by an algorithm which extracts similar parts of objects from reconstructed the point cloud of a scene and generates an initial grasping pose on the part, e.g., [15].

Starting at an initial grasp with an injected pose error, our grasp adjustment method began to adjust the hand. When the robot had exited the pipeline of Figure 2 via state *Stable Grasp Achieved* of Figure 2, it would lift up the object. After the lift up action, a "shake test" took place by rotating the last joint of the robotic arm within a range of ± 60 degrees. The scoring criteria for a grasping test were as follows: if the object falls on the table after lift up or the shake test, score 0; if the object moves in hand during the motion of finger close, lift up, or the shake test but stays in hand in the end, score 0.5; if the object stays stable in hand throughout the entire grasping process, score 1. The intuition behind this set of criteria was that the object should remain stable during finger close, lift up, and a shake test to maximally preserve the static status of the object.

Table I provides detailed information concerning the performance of our grasping pipeline. As a comparison, we also ran a conventional grasping pipeline starting at each of these 10 initial poses but without the post-execution grasp adjustment procedure. Figure 7 summarizes the comparison between our grasping pipeline and a conventional grasping pipeline without a post-execution grasp adjustment procedure. It is shown that our grasp adjustment procedure increases the grasping performance compared to a conventional grasping pipeline where no grasp adjustment is exploited.

TABLE I

DETAILS OF GRASPING NOVEL OBJECTS WITH A TACTILE EXPERIENCE DATABASE

Object	# of grasps	Avg. # adj.	Lift-up	Score
Box	10	2.0	9	0.9
Paint bottle	10	1.3	10	1.0
Wineglass	10	1.2	10	1.0
Mug	10	1.0	10	0.95
Canteen	10	1.2	9	0.8

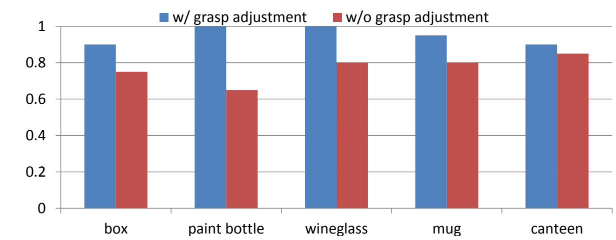


Fig. 7. Performance on grasping novel objects w/ and w/o our post-execution grasp adjustment procedure. Bars in blue show the scores of grasping using our method and bars in red show the scores of grasping without the grasp adjustment procedure.

VII. DISCUSSION

To extend our previous work to novel objects, we selected shape primitives to sample potential similar local geometries where stable grasps can be established and used their tactile experience to guide a grasping process. Although the grasps sampled on shape primitives can be used on a wide range objects, it is still important to seek for other alternative approaches to the parameterization of local geometry and construct our tactile experience database. One option would be utilizing the Columbia Grasp Database (CGDB) [16], which contains over 200,000 stable grasps on about 8,000 object models using several different robot hand models, including a Barrett hand, a PR2 gripper, and a human hand model. Statistically, the stable grasps in the CGDB should sample a wider range of local geometries. We are currently working on constructing a tactile experience database using stable grasps from the CGDB and evaluating the performance of using this tactile experience database to grasp novel objects under pose uncertainty.

In this paper, we manually chose the initial grasping pose for each test object based on geometric similarity. As part of our future work, we will be designing algorithms which synthesize grasp candidates based on reconstructed local geometries. With such a grasp generator, our grasping pipeline could be further automated.

We are also looking at different sensors to integrate into our method. Currently, we have used tactile sensors to indicate active contact. However, one practical issue of using tactile sensors is that they may not be sensitive enough to prevent objects from being knocked down due to contact forces. The introduction of unexpected disturbance to object pose will bring difficulty into a grasp adjustment process. We think proximity sensors would alleviate this issue since potential touch can be predicted before contacts are established.

VIII. CONCLUSION

In this paper, we focus on developing a grasp adjustment process for grasping novel objects with similar local geometry. Our goal is to improve grasping performance starting with an unstable grasping pose due to pose uncertainty. This paper is based on our previous work where we used precomputed tactile experience to guide a grasp adjustment process on known objects [1]. To extend our method to novel objects, we selected a series of shape primitives to parameterize similar local geometries that could be shared by different objects and built a new tactile experience database. Using this tactile experience database, we were able to synthesize effective hand adjustments which improve the grasping performance of novel objects under pose uncertainty.

REFERENCES

[1] H. Dang and P. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*. in press, available at http://www.cs.columbia.edu/~dang/tr2013.pdf.

[2] D. Berenson, S. Srinivasa, and J. Kuffner, "Addressing pose uncertainty in manipulation planning using task space regions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, October 2009.

[3] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Robotics and Automation, 2011 IEEE International Conference on*, pp. 2851 –2858, may 2011.

[4] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal, "Learning to grasp under uncertainty," in *Robotics and Automation, 2011 IEEE International Conference on*, pp. 5703 –5708, may 2011.

[5] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *Robotics and Automation, 2012 IEEE International Conference on*, pp. 557 –562, may 2012.

[6] J. Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, and N. S. Pollard, "Physically-based grasp quality evaluation under uncertainty," in *Robotics and Automation, 2012 IEEE International Conference on*, pp. 3258 –3263, may 2012.

[7] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," in *Robotics: Science and Systems (RSS 2007) Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

[8] R. Platt, A. Fagg, and R. Grupen, "Null-space grasp control: Theory and experiments," *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 282–295, 2010.

[9] K. Hsiao, L. Kaelbling, and T. Lozano-Prez, "Robust grasping under object pose uncertainty," *Autonomous Robots*, vol. 31, pp. 253–268, 2011.

[10] J. Laaksonen, E. Nikandrova, and V. Kyrki, "Probabilistic sensor-based grasping," in *Intelligent Robots and Systems, 2012 IEEE/RSJ International Conference on*, pp. 2019–2026, 2012.

[11] H. Dang and P. Allen, "Learning grasp stability," in *Robotics and Automation, IEEE International Conference on*, pp. 2392–2397, 2012.

[12] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++." [web page] http://www.ics.forth.gr/~lourakis/levmar/.

[13] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[14] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.

[15] R. Detry, C. Ek, M. Madry, J. Piater, and D. Kragic, "Generalizing grasps across partly similar objects," in *Robotics and Automation, 2012 IEEE International Conference on*, pp. 3791 –3797, may 2012.

[16] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen, "The columbia grasp database," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 1710 –1716, may 2009.