Planning Visual-Tactile Precision Grasps via Complementary Use of Vision and Touch

Martin Matak[1] and Tucker Hermans[1,2]

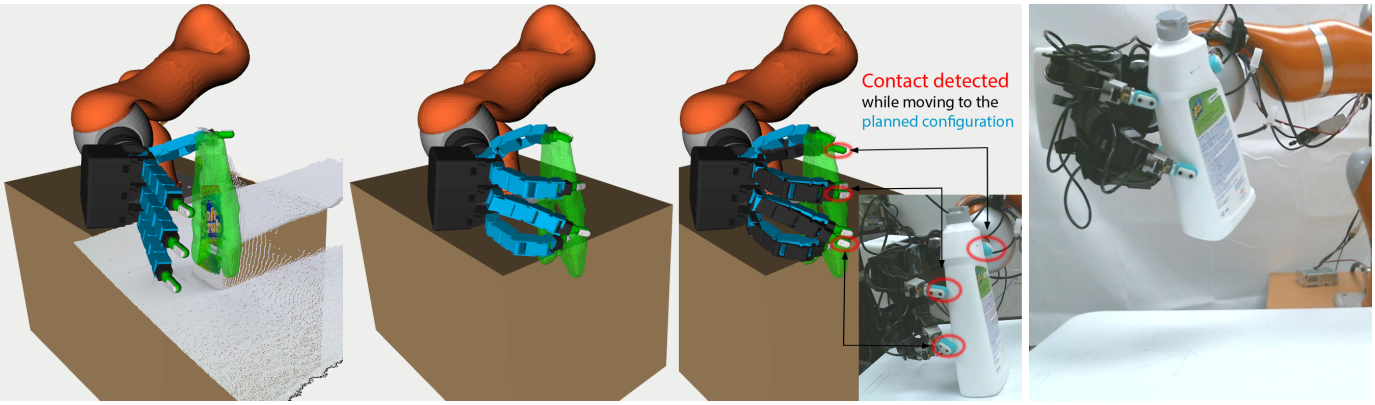

Fig. 1: Our approach operates in four phases. (1) The robot plans a grasp preshape using a learned model (left). (2) The robot plans a grasp where its fingertips contact the estimated object surface (center left). (3) The robot executes the planned grasp (shown in blue) adapting based on tactile sensing to account for inaccuracies in the reconstruction (center right). (4) The robot lifts the object if its in-contact grasp classifier predicts success (right).

***Abstract*—Reliably planning fingertip grasps for multi-fingered hands lies as a key challenge for many tasks including tool use, insertion, and dexterous in-hand manipulation. This task becomes even more difficult when the robot lacks an accurate model of the object to be grasped. Tactile sensing offers a promising approach to account for uncertainties in object shape. However, current robotic hands tend to lack full tactile coverage. As such, a problem arises of how to plan and execute grasps for multi-fingered hands such that contact is made with the area covered by the tactile sensors. To address this issue, we propose an approach to grasp planning that explicitly reasons about where the fingertips should contact the estimated object surface while maximizing the probability of grasp success. Key to our method's success is the use of visual surface estimation for initial planning to encode the contact constraint. The robot then executes this plan using a tactile-feedback controller that enables the robot to adapt to online estimates of the object's surface to correct for errors in the initial plan. Importantly, the robot never explicitly integrates object pose or surface estimates between visual and tactile sensing, instead it uses the two modalities in complementary ways. Vision guides the robots motion prior to contact; touch updates the plan when contact occurs differently than predicted from vision. We show that our method successfully synthesises and executes precision grasps for previously unseen objects using surface estimates from a single camera view. Further, our approach outperforms a state of the art multi-fingered grasp planner, while also beating several baselines we propose.**

***Index Terms*—Multifingered Hands; Grasping; Deep Learning in Grasping and Manipulation**

I. Introduction

Manuscript received: September 2, 2022; Revised November 18, 2022; Accepted December 14, 2022.

This paper was recommended for publication by Hong Liu upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by supported by NSF Award #1846341, by DARPA grant N66001-19-2-4035, and by a Sloan Research Fellowship.

[1] Martin Matak and Tucker Hermans are with the School of Computing and the Robotics Center, University of Utah, Salt Lake City, UT, USA. martin.matak@utah.edu [2] NVIDIA; Seattle, WA, USA

Digital Object Identifier (DOI): see top of this page.

In this letter we look into enabling a robot to pick up everyday objects that the robot hasn't encountered before. We focus on the task of grasping previously unseen objects as a fundamental step toward successfully deploying robots in household environments. In such environments, the robot should be able not to only pick up objects, but also to use them for everyday tasks [1]. This requires the ability of fine in-hand manipulation of objects, which makes multi-fingered hands a good candidate as a general end effector. While multi-fingered hands seem promising for downstream tasks, using them to grasp everyday objects and ensuring the resulting grasp makes contact only with the fingertips remains a challenging problem [2,3].

It's obvious that contact with an object must be made in order to successfully pick the object up, but it's unclear how to explicitly enforce this constraint when a robot doesn't know the complete object shape. This is the core focus of this paper—establishing an in contact, precision grasp with a previously unseen object when given a single RGBD image from a fixed camera-view. To help account for the lack of information a single camera view provides, we examine how tactile sensing can improve the grasp control process. In particular we examine how to do this when the robot only has tactile sensing on its distal links (fingertips), a common setup on contemporary hands [4–6].

In general, tactile sensors improve grasping. Calandra and colleagues [7] show that tactile feedback, in addition to vision-based input, provides more information for computing a good grasp, while Murali et al. [8] present an approach to grasping using tactile sensing without vision. However, it's unclear how to get to a grasp such that tactile feedback is registered, i.e. the contact is made with the tactile sensors when planning with fully actuated, dexterous hands. Having only limited tactile coverage on most robotic hands doesn't help either. One approach [2] is to explicitly model a *precision*

grasp, which would result in a grasp where contact with the object is made with fingertips of the hand, exactly where the tactile sensors are. After the hand is in contact with the object, Veiga et al. [9] show how to stabilize the grasp using tactile feedback. Apart from the tactile-guided approaches, there are many vision-based methods to tackle grasping [10]. Several of these use 3D reconstruction from partial views to aid the grasp planning process [11–13] and others examine reconstructing object shape from touch [8,14,15]. However, no work has shown the ability to plan, adapt, and ensure precision grasps for multi-fingered hands through joint visual and tactile sensing in the real world. Indeed, as discussed in Section II, most multi-fingered grasping approaches use simple, heuristic controllers to move from a grasp preshape to an in-contact grasp. Further, similar works evaluated in simulation make unrealistic assumptions such as full tactile coverage of the robot and the ability to repeatedly make contact with the object without it moving.

To address this gap, we propose a learning-guided and geometry informed grasp controller to autonomously generate precision grasps on previously unseen objects. We formulate grasp synthesis as an optimization problem with a hard constraint that contact must be made between the estimated object surface and the robot's tactile fingertips. The robot uses a neural network that encodes the probability of grasp success as its objective function as in [16]. We generate an initial estimate of the object surface from the partial-view point cloud using the PointSDF [11] neural network.

While the estimated object surface and pose provide a model useful for planning a grasp, we also know it is very likely inaccurate. To account for this we leverage tactile feedback during the grasp control phase to adapt to the uncertain object shape and ensure contact. Notably, we do not fuse the visual and tactile surface estimates into a single model, instead we use them to separately plan and ensure contact. As such we avoid the difficult problem of registering the different measurements under extrinsic calibration error and possible object motion. After closing the hand, the robot evaluates an in-contact grasp classifier to decide whether to lift the object. Figure 1 visualizes all stages of our approach.

We experimentally demonstrate that our approach can reliably generate multi-fingered precision grasps for novel objects without explicitly integrating vision and touch. We run real world robot experiments that quantify that using both vision and touch improves grasp performance over using vision alone. Our results additionally characterize the benefits of our approach over prior work. We successfully plan and execute precision grasps for various, novel objects observed from a single camera view. Some examples of our grasps are shown in Fig. 4. Experiments videos and source code can be found on our project website: https://sites.google.com/view/precision-grasps.

II. Related work

We divide our discussion of previous grasp planning work between geometry-based and learning-based approaches before discussing grasping with tactile sensing.

A. Geometry-based Grasp Synthesis

Geometric-based grasping approaches leverage a model of the object geometry in order to compute a grasp metric. One of the most popular approaches in this category is Eigengrasp [17]. Eigengrasp computes a grasp pose and joint configuration in a low-dimensional "eigengrasp" space and closes the fingers at equal joint velocity rates until detecting contact. Assuming full knowledge of object shape, Roa and Suarez [18] determine contact regions on the object to make contact with, such that the resulting grasp is in force closure. Rosales et al. [19] solve for hand configuration to be in contact with pre-specified regions on the object. Zheng and Qian [20] quantify friction and contact position uncertainty when computing force closure grasps.

Another geometry-based, uncertainty aware grasp synthesis approach comes from [21], which fuses multiple camera view to infer a "pre-touch" configuration to maximize grasp success. Hang et al. [22] synthesize a grasp that is in contact with an object, and once the hand is in contact, use a learned probabilistic controller to stabilize it. Siddiqui [23] use a Bayesian inspired method to efficiently explore a variety of grasps. After moving to a predicted pose, the hand is closed using a simple controller until contact is detected. Morales et al. [24] build a database of possible power and precision grasps and retrieve a grasp for a target object at run time. They obtain precision grasps using a simple heuristic controller, but execute no grasps in the real world. Planning to grasp an object with uncertain pose, but known geometry, has also been examined formally as a belief-space planning problem [25], however, scaling this to be efficient with uncertainty in shape, while likely useful, is an open challenge. In the case of uncertain shape, such as when estimated from a single camera view as is our focus, these geometry-based grasp metrics will likely be inaccurate or suboptimal due to inaccuracies in estimated shape.

B. Learning-based Grasp Synthesis

The research literature contains numerous data-driven grasp approaches [3,7,26–30]. We build on the approach from [16] to generate a grasp preshape. The primary novelty of our work comes after the robot moves to this preshape. We classify learning-based grasp synthesis approaches between those that generate grasps in contact with the object and grasps that are close to, but not in contact with the object. We found no work that combines the two.

In contact grasps: Varley et al. [27] train a CNN to predict heatmaps for fingertip locations on the object as well as palm pose, which are used to synthesize a grasp. While Veres et al. [31] generate contact points on an object for a 3-fingered hand using generative models (in simulation), Shao et al. [32] manage to synthesize grasps as contact points on an object for various grippers and demonstrate their approach in a real-world setting. Sundermeyer and colleagues [33] predict grasp locations such that one finger of a parallel jaw gripper makes contact with the observed point cloud, this allows for a clever reparameterization of the 6DOF prediction to 4DOF. However the second finger location is determined based on a predicted gripper width, so it is unclear how to scale this approach to

fully actuated multi-fingered hands. All approaches in this category predict grasps in contact with the object and are therefore brittle to shape estimation and camera noise. To resolve this, other approaches predict grasps near the object, but not in contact.

Near contact grasps: Kappler et al. [26] predict grasp as a palm pose, Wu et al. [34] use a policy gradient method to predict grasp pose and joint angles from a depth image. Lundell and colleagues [12,35] use a GAN to predict grasps that are almost on the surface of the object, yet not in contact. Similar to [3], Mousavian et al. [36] initialize and refine a grasp pose using a learned neural net. Our own previous work [2,3,11,16] synthesizes a preshape similarly as in this letter, but doesn't explicitly reason about making contact either through vision or touch. All the approaches in this category after moving the robot to the preshape use either a simple heuristic controller to close the hand or don't specify how exactly the hand is closed. We identify this as a gap in the literature and explore it here. For a deeper overview of dexterous grasping, we refer the reader to [37].

C. Tactile Enabled Grasp Evaluation and Control

Calandra and colleagues [38] propose a grasping approach that maximizes grasp success probability using a sampling based method. Their model predicts success probability based on tactile readings, RGB image, and a regrasping action for a two-fingered gripper. If the predicted grasp success probability is above a threshold (90%), the robot attempts to lift the object. Hogan et al. [5] follow a similar approach but use a tactile-only grasp quality metric. For their resampling strategy they simulate various actions (movements of the gripper) and pick the action with the highest probability of success, as predicted by the tactile-only metric. Wu et al. [39] use a deep reinforcement learning method to implement a closed-loop approach to grasping. They use tactile sensors on the 3-fingered Barett Hand as part of the observations used by the RL policy. Dang and Allen [40] regrasp by searching for the closest stable grasp in the database of all the previous grasps performed by the robot. After finding such a grasp, they adjust the hand. Similar to the rest of the work in this section, the tactile readings are used for the grasp quality metric, but also to find similar grasps in the database. Murali et al. [8] use learned corrective actions that map tactile input in latent space to gripper adjustment in task space. Dragiev et al. [14] develop a tactile-informed regrasp control law. Farias et al. [41] estimate object shape from vision, update the shape estimate using tactile feedback, and use the resulting model to estimate force closure. Both papers [14,41] only evaluate in simulation and assume the robot can detect contact anywhere on the hand.

III. Problem Description

The problem we wish to solve has two main phases. First, the robot plans a precision grasp that maximizes the probability of grasp success under uncertain object surface estimation. Then, the robot must execute the planned grasp using tactile feedback to compensate for shape uncertainty.

Let $f(\boldsymbol{q}, \boldsymbol{o})$ estimate the probability of grasp success for configuration $\boldsymbol{q}$ for object $\boldsymbol{o}$. Let the robot have n degrees of freedom (DoF) in the arm and m DoF in the hand. We are searching for the robot configuration $\boldsymbol{q} \in \mathbb{R}^{n+m}$ such that every fingertip is on the surface of the object $\boldsymbol{o}$ and that f is maximized. The surface of the object $\boldsymbol{o}$ is estimated from a partial pointcloud. Let $\phi_p^i : \mathbb{R}^4 \rightarrow \mathbb{R}^3$ denote the forward kinematics function for the position of the central point on the surface of the fingertip i. Input to ϕ_p^i is in $\mathbb{R}^4$ because every finger on the hand we work with has 4 DoF. Let $\texttt{SDF}(p, \boldsymbol{o})$ be the signed distance function between a point $p \in \mathbb{R}^3$ and the object $\boldsymbol{o}$, returning positive values for p outside the object, negative inside, and 0 on the surface.

We formalize the search for the grasp configuration $\boldsymbol{q}^*$ as

$$\boldsymbol{q}^* = \arg\max_{\boldsymbol{q}} \quad f(\boldsymbol{q}, \boldsymbol{o}) \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{q}_{min} \preceq \boldsymbol{q} \preceq \boldsymbol{q}_{max} \tag{2}$$

$$\texttt{self\_coll}(\boldsymbol{q}) = 0 \tag{3}$$

$$\texttt{SDF}(\phi_p^i(\boldsymbol{q}), \boldsymbol{o}) = 0 \; \forall i \in \{1, \ldots, F\}. \tag{4}$$

Equation (2) constrains the solution within joint limits, while the function $\texttt{self\_coll}(\cdot)$ counts the number of hand links in self-collision and F denotes the number of fingers we wish to make contact with the object. In our experiments $F = 4$.

A solution to the optimization problem defines a grasp such that fingertips are in contact with the object surface and not in self collision. However, the estimated object shape is likely imperfect and hence there is inherently a mismatch between estimated object surface and the real one. This defines the second part of the problem - moving the robot to the planned grasp knowing a contact with the object might happen before or after expected. Not accounting for this may result in knocking the object over or stopping before contact is made. Formally, given current configuration $\boldsymbol{q}[k]$, tactile readings $\tau[k]$, and target configuration $\boldsymbol{q}_F$, the robot must execute a control policy $\boldsymbol{u} = \pi(\boldsymbol{q}[k], \tau[k], \boldsymbol{q}_F)$ to achieve a grasp where its tactile fingertips contact the object.

IV. Our Approach

Our approach consists of four steps: (1) planning and moving the robot arm and hand to a grasp preshape configuration, (2) planning an in-contact grasp using the estimated object shape, (3) moving the robot fingers to the planned in-contact grasp configuration and (4) evaluating the robot's in-contact classifier to decide whether to lift the object or regrasp it. The first two steps base decisions purely on point cloud input, whereas the third step uses tactile feedback to adapt the plan online. The fourth step uses visual input and the current robot configuration. We do not fuse visual and tactile sensing to make a unified model of the object surface, instead we use the two separately to satisfy the contact constraint during planning and control respectively. Figure 1 visualizes these steps. We now describe the four components in turn, followed by implementation details of our learned models.

A. Grasp Preshape Synthesis

As the first step in our approach we plan a grasp preshape $\boldsymbol{q}_0 = [\boldsymbol{q}_A, \boldsymbol{q}_H]$ which maximizes the probability of grasp

success, but does not make contact with the object. The preshape contains both the arm configuration $\boldsymbol{q}_A \in \mathbb{R}^n$ and the hand configuration $\boldsymbol{q}_H \in \mathbb{R}^m$. We solve for the preshape using the approach from [16] which maximizes the probability of grasp success using a learned neural network as its objective. The objective contains two terms: a mixture density network (MDN) as a prior, and a preshape classifier as a likelihood component. The MDN additionally serves as a sampler to initialize the optimization. Both neural networks take as input a voxelized representation of the object from the observed single-view RGBD image. We give further details about the networks and their training below. The preshape planner serves to move the palm of the hand to a pose close to the object and the proximal joints of each finger into a configuration amenable to making a high quality precision grasp. As in [16] we optimize using bound constrained BFGS [42]. Figure 1 left shows an example grasp preshape.

B. In-Contact Grasp Synthesis and Evaluation

After moving to the preshape configuration $\boldsymbol{q}_0$, the robot plans to make contact with the object. The robot uses $\boldsymbol{q}_0$ to initialize its optimization for a hand configuration in contact with the object. This differs from the previous step, where we optimize for arm and hand configuration. For the rest of the approach, the arm configuration $\boldsymbol{q}_A$ remains fixed and we only change the joints of the fingers $\boldsymbol{q}_H$.

We reformulate the in-contact grasp synthesis problem from Eqs. (1)-(4), by relaxing the SDF constraint and using an estimate of the SDF, $\widehat{\texttt{SDF}}$. We add an additional term to the objective to encourage alignment between the fingertip surface normal and the approach vector resulting in the following penalty-method objective, $\tilde{f}(\boldsymbol{q}, \boldsymbol{o}, \boldsymbol{p}) =$

$$\sum_{i=1}^{F} ||\widehat{\texttt{SDF}}(\boldsymbol{p}_i, \boldsymbol{o})||^2 + ||\phi_p^i(\boldsymbol{q}) - \boldsymbol{p}_i||^2 + \alpha||1 - \cos\theta_i||^2 \quad (5)$$

where θ_i is angle between the normal on ith fingertip center point and the approach vector $\boldsymbol{p}_i - \phi_p^i(\boldsymbol{q})$. Here $\boldsymbol{p}_i$ serves as an auxiliary decision variable defining a point between the fingertip and object as in [43]. We initialize it halfway between the ith fingertip and estimated object surface. We found that including $\boldsymbol{p}_i$ was important for numerical stability. Finally, α weighs between minimizing distance and aligning orientation of the fingertip. We describe implementation details of $\widehat{\texttt{SDF}}(\cdot)$ later in this section.

We solve the objective from Eq. (5) using a Gauss-Newton solver with bound constrains to handle the joint limits with $\alpha = \frac{1}{j^2}$. Here j defines the current solver iteration. We further implement a projection method on our line search to handle self collisions of the hand. We show an example of a solution in Fig. 1 center left. We can see that the planned configuration makes contact with the object at the estimated object surface. However, the estimated object surface is likely inaccurate due to error in object pose estimation and shape reconstruction. In the next subsection, we describe how we execute our planned grasp to handle this uncertainty.

C. Making Contact

To move to the target configuration, we compute time-optimal trajectories for each finger [44] and then scale the trajectories to be of equal duration, so that all fingers reach their target configurations at the same time. This lowers the chance of moving the object when making contact. The robot evaluates the constraint described in Eq. (4) online using tactile feedback during trajectory execution. This allows the robot to detect contact before the trajectory is completely executed. If the tactile classifier $t(\tau[k])$ detects contact at timestep k, the constraint is satisfied and the finger in contact stops moving. If after K steps a finger reaches the desired configuration, but does not detect contact, the robot moves its finger toward the object surface following the gradient of $\widehat{\texttt{SDF}}$ in attempt to make contact. The finger stops following the $\widehat{\texttt{SDF}}$ gradient if it hasn't made contact after $\bar{k}$ additional control iterations as this implies the shape estimate gradient is uninformative at this location. Formally, $\pi(\boldsymbol{q}[k], \tau[k]) =$

$$\begin{cases} 0 & \text{if } t(\tau[k]) \geq p_c \vee k > K + \bar{k} \\ \Delta\boldsymbol{q}[k] & \text{if } t(\tau[k]) < p_c \wedge k \leq K \\ -\gamma \nabla_{\boldsymbol{q}} \widehat{\texttt{SDF}}(\phi_p(\boldsymbol{q}), \boldsymbol{o}) & \text{if } t(\tau[k]) < p_c \wedge K < k \leq K + \bar{k} \end{cases} \quad (6)$$

where $\Delta\boldsymbol{q}[k]$ is the initial plan of K steps; p_c is the tactile pressure threshold of 365 $\texttt{Pa}$; $\gamma = 1.0$; and $\bar{k} = 1$.

The hand tracks the joint positions commanded by π using an inverse-dynamics torque controller. An example of a grasp after running our controller can be seen in Fig. 1 center right. Details about the classifier $t(\cdot)$ are explained in the next subsection. After making contact, we increase grasp force prior to lifting using a heuristic. If more force is not applied, the object slips through while the arm is attempting to lift it. The heuristic commands 0.7 $\texttt{rad}$ increase to all but the first joint on non-thumb fingers, and 0.5 $\texttt{rad}$ increase to last two joints of the thumb. A promising direction for future work is a more systematic reasoning about the applied force.

Despite making contact with the object surface, the resulting configuration might not generate a successful grasp. To estimate whether the robot should lift the object with the resulting configuration, we evaluate our in-contact grasp classifier h_c to estimate the probability of grasp success. When evaluating h_c, we assume the object pose has not changed, despite applying more force, and feed in the same visualize input as for the preshape classifier. In-hand object pose tracking is a challenging problem and out of the scope for this paper. Similar to previous works [7,39], if the output of the in-contact grasp classifier is below a threshold, the robot can reject the grasp and attempt a regrasp.

D. Learned Model Architectures

We now describe the learned MDN $g(\cdot)$ and the preshape grasp classifier $h_p(\cdot, \cdot)$, which are used to generate the preshape solution $\boldsymbol{q}_0$. Then we describe the model used to estimate the initial object shape $\widehat{\texttt{SDF}}(\cdot)$ and the tactile classifier $t(\cdot)$ used to detect contact with the object. Finally, we describe our in-contact grasp classifier $h_c(\cdot, \cdot)$ which is used to decide whether to lift the object or attempt a regrasp.

Following [16], we use a mixture density network (MDN) $g(\cdot)$ that takes the voxelized object representation as an input and outputs a palm pose in object frame, as well as the joint positions for the two joints closest to the palm for each finger.

We model the preshape grasp classifier $h_p(\cdot,\cdot)$ as a neural network described in [16]. The classifier receives as input the palm pose in object frame, the voxelized object representation, the object dimensions and the first two joint values of every finger. The classifier outputs estimated probability of grasp success. Further details on the training procedure are in Section V. We model our in-contact grasp classifier h_c similarly to our preshape grasp classifier h_p with one difference: h_c receives all hand joints as input, while h_p receives only two joints per finger.

To reconstruct object shape from a single view, we use the PointSDF model from [11] as our implementation of $\widehat{\texttt{SDF}}$. The model takes as input the object point cloud and a query point and outputs the signed distance to the query point, scaled to between -1 and 1. For further details see [11].

We use the BioTac sensor installed on all fingertips for tactile feedback [45]. We define a contact classifier $t(\cdot)$ that receives as input the fluid pressure in the sensor (P_{DC}). Before the hand starts to close, the robot tares the sensor, such that the current pressure value reads zero. When closing the hand, if the pressure exceeds the preset threshold p_c for 10 consecutive timestamps, the classifier $t(\cdot)$ reports contact. The sensor is sampled at a frequency of 100 $\texttt{Hz}$.

V. Experiments

We start this section by describing the data collection procedure used to train the grasp classifier. We then explain our baselines for comparison before discussing experiments and results for evaluating making contact and grasp success. Our experiments show: (1) that our proposed approach outperforms the state-of-the-art (SoTA) comparison as well as tactile enhanced SoTA; (2) the benefits of having both preshape and in-hand classifiers; and (3) the importance of estimating object geometry through the learned SDF. Figure 4 shows example grasps generated by our approach.

A. Simulation: Data Collection and Training

We conduct all training only in simulation using the four-fingered, $m = 16$ DoF Allegro hand mounted on a Kuka LBR4 $n = 7$ DoF arm, which is the same as our real world setup. We use a Microsoft Azure Kinect camera to generate the point cloud of the object on the table. We collected simulated grasp data using our robot hand-arm setup inside the Gazebo simulator with the DART physics engine. We use the built-in Gazebo Kinect camera to generate point clouds simulating the Microsoft Azure Kinect camera we use in real-world experiments.

We collected training data using a heuristic, geometry-based planner adapted from [16]. However, our approach differs in how the robot closes its hand after moving to the preshape. We formulate hand-closing problem as the optimization problem defined in Eq. (5), where we set $\alpha = 0.0$ to speed up the optimization, and solve it using Gauss-Newton. After moving to the computed joint position, we increase the stiffness of the grasp by $0.2\texttt{rad}$ to mimic a grasp stability controller. During data collection for training (and only for training), we use the true mesh of the object and no tactile feedback. To check for self-collisions, we use the GJK algorithm [46]. In total, we collected 13,831 samples, out of which 3777 are successful grasps.

We pretrain a voxel autoencoder on a 3D reconstruction task [16] and extract the voxel encoder from the network for use in the grasp classifiers h_p and h_c. We freeze the extracted weights and train the rest of the classifier using the Adam optimizer. We find that pretraining the autoencoder was a necessary step to train the classifiers successfully. On the validation set (10% of the collected dataset) h_p and h_c achieve 78% and 79% accuracy respectively. This is comparable to numbers reported in [16]. Following [16], we fit a mixture density network (MDN) that takes the object point cloud as input and outputs a grap preshape sample. The MDN uses the same voxel encoder as the classifiers.

Fig. 2: The objects used in experiments. Objects from left to right; back row: "pitcher", "soft scrub", "pringles", "wooden block", "cheezeit", "master chef", "domino", "mustard"; front row: "bowl", "red cup", "yellow cup", "squishy brick", "rubic cube", "tuna can", "jello red", "spam", "jello brown" and "chain".

B. Baselines for Comparison

We implement five baseline comparisons to quantify the benefits of the different components of our approach. Our first baseline, **B1** uses the heuristic, grasp controller from [2] to close the hand. This represents a SoTA comparison to our complete system. Since **B1** does not use tactile sensors, we implement **B2** that uses the same controller as **B1**, but stops when the tactile classifier detects contact instead of the sensed joint velocities. This allows us to understand to what extent any improvement comes from adding tactile sensors to the grasp controller. Baseline **B3** samples a preshape configuration from an MDN trained on all positive samples from the training dataset, but closes the hand using our visual-tactile controller. This examines whether our controller removes the need for the initial preshape optimization and hence the preshape classifier. Further, we look into the importance of object shape completion by introducing **B4** which reconstructs only the visible region of the object using marching cubes [48] from the camera's partial point cloud, which we use with the trimesh library [49] to evaluate the SDF. This contrasts with our approach which uses PointSDF to estimate the entire shape

of the object. We implement baseline **B5** as an additional comparison for evaluating the ability for the controllers to make fingertip contact with the object without moving it. We simply execute our controller π open-loop stopping after K steps ignoring tactile feedback.

C. Making Contact in the Real World

We first evaluate our proposed controller on generating grasps that make fingertip contact with the object without moving it. We report the number of fingertips in contact with the object after closing the hand, as well as whether the object remained flat on the table, tilted, or was knocked over by the robot. We compare our controller to **B1** and **B5** by generating and saving a preshape for a given object using our grasp planner and then executing the selected grasp controller. We manually label whether the object tilted or fell. We run experiments on 5 different objects (pringles, domino, soft scrub, mustard, and wooden block) in 5 different poses resulting in 75 hand closing episodes in total.

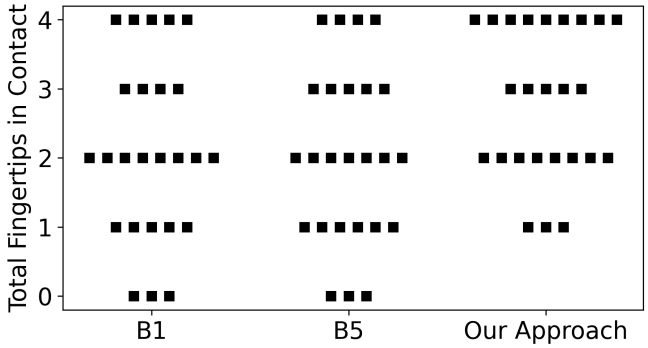

Fig. 3: Number of fingertips in contact with the object after closing.

Figure 3 visualizes our results. Our approach results in more fingertips in contact with the object than using **B1** or **B5**. The object tilted only 3 times when using our controller, while **B5** tilted the object 10 times, and **B1** tilted 21 times. Further, we report that our method never knocked over the object, while the vision-only **B5** knocked over an object once, and **B1** did so three times. Our method sometimes failed in making all four fingers contact the object. We believe three issues contribute to this: (1) the object simply stands out of reach for some fingers for the generated preshape, (2) the $\widehat{\text{SDF}}$ is not accurate enough to lead the controller in the right direction, (3) the tactile contact classifier is sometimes inaccurate. While imperfect, we show next that our approach provides advantages over the comparison methods for our task of interest–grasping.

D. Grasping in the Real World

We perform real-world experiments using 18 objects from the YCB [47] object set. These objects span different textures, shapes, masses, and sizes. We show the experimental setup and objects used in Fig. 2. All experimental objects but the pringles can are unseen in training. For experiments, we place a single target object on the table and run our pipeline. We chose a fixed test location on the table for the target object and rotate the object to 3 different angles, keeping the poses constant across the methods and objects. Given a resulting robot joint configuration from our grasp planner, we perform motion planning using the RRTConnect planner from MoveIt! to obtain a collision free trajectory to the preshape configuration. We provide the bounding box of the object to the motion planner for collision checking. We find that this conservative approach yields fewer collisions with the object compared to providing the reconstructed mesh of the object to the motion planner. After executing the motion plan, the robot closes its hand using the grasp controller. Then, the robot attempts to lift the object approximately 15cm above the table. Only if the object is in the hand after executing the lifting trajectory, we label the attempt as a successful grasp. If the robot moved to the preshape, but failed to lift the object, we label the attempt as unsuccessful. The robot attempts 270 grasps total, 54 for each of the 5 comparative methods.

Fig. 4: Example precision grasps resulting from our controller.

1) Quantitative results: Figure 5 presents the grasp success rates for the taller objects that comprise the back row in Fig. 2. Overall success rates for **B1-4** and our approach are $65\%, 28\%, 46\%, 75\%$, and 75%, respectively.

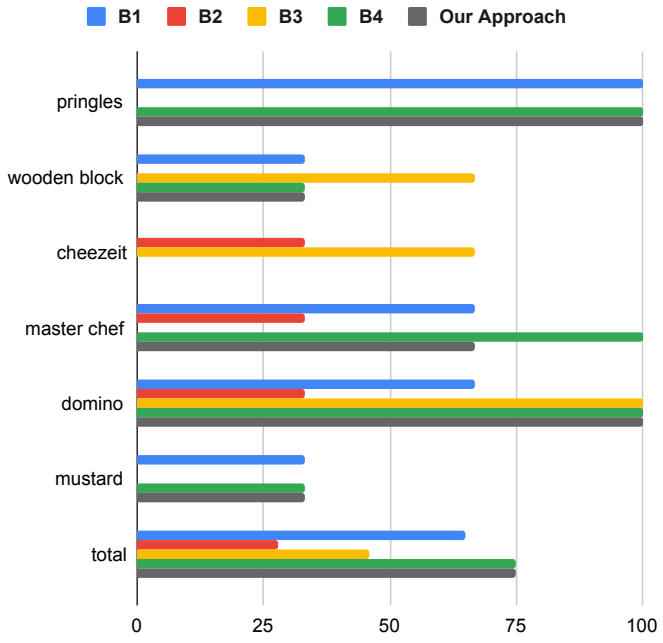

Fig. 5: Grasp success rates for each tall object and overall.

Our approach outperforms most of the baselines and successfully picks up most of the objects. However, **B1**, **B4**, and our approach all fail to find a plan to pick up "cheezeit". This arises from the initial grasp preshape optimization always coming into contact with the object. This seems to be somewhat an issue of random initialization as **B2**, which uses the same preshape generation was successful once. However, both **B2** and **B3** had the overall lowest performance. Only **B4** has a success rate equal to our proposed approach. As such we see that the proposed in-contact grasp planner and associated tactile feedback controller combine to improve over the previous SoTA **B1**. We can attribute the lack of improvement from using tactile feedback in **B2** to the fact that the heuristic grasp controller does not explicitly try to make contact with the tactile fingertips. Thus **B2** would often make contact with other parts of the hand when closing pushing away or knocking over the object. In contrast, while our in-contact planner would generate target configurations that penetrate the object, the resulting grasps often succeeded as the robot detected contact with the tactile sensors.

Recall that the only difference between our proposed approach and **B4** is the SDF estimate. To further investigate the difference in performance, we examine the "low profile" objects shown in the front row of Fig. 2. We define "low profile" objects to be objects shorter than 10cm. The tallest low-profile object is "jello brown" with a height of 89mm. Our approach achieves a **40%** rate for low-profile objects, while **B1-4** have respective success rates of $20\%, 9\%, 10\%$ and 18%. This result demonstrates a substantial improvement in using PointSDF for reasoning about small-object geometry compared with running marching cubes in **B4** for planning.

That said, across the 10 different low profile objects, we were able to execute only 78 out of 150 attempts, because of the resulting preshape coming into collision with the table. This arises from the lack of explicit reasoning about collisions with the environment in our optimization. We point out that this is one of the biggest shortcomings of our method and we believe this is a promising avenue for future work.

Finally, we look into the performance of our in-contact grasp classifier h_c. We analyze the performance of the classifier using only **B3, B4** and our method as those use our proposed grasp controller. We filter out the data where the trajectory couldn't be executed. We show the resulting precision-recall curve for executed grasps in Fig. 6. This shows the classifier performs well in the real world on previously unseen objects despite being trained only in simulation. It also shows that the classifier can be used to further increase grasp success rates by executing grasps only if the classifier prediction is above a high threshold. To construct the precision-recall curves, in our experiments we didn't filter out any grasps based on the estimated probability of success. Finally, we see that the h_c obtains both high precision and recall when coupled with our proposed approach.

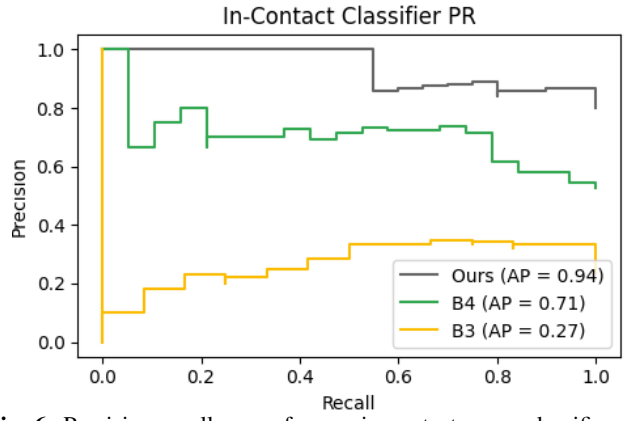

Fig. 6: Precision-recall curve for our in-contact grasp classifier, h_c. We compare the performance of our approach, **B3**, and **B4**, the three approaches that use our proposed tactile controller.

2) Qualitative results: While our approach outperforms the baselines quantitatively, we think it shines brightest when one looks at the generated grasps. In Fig. 4 we show some grasps resulting from our controller, where the robot successfully grasps objects such that the tactile sensors contact the object. Since **B1** and **B2** don't explicitly reason about making contact with the fingertips, we found that the resulting grasps tend to make contact with other parts of the hand instead. This makes our approach suitable for downstream tasks that require updating the object shape or pose estimate or requiring precision in-hand manipulation.

VI. Conclusion

We proposed an approach for a robot to plan and execute multi-fingered, precision grasps for previously unseen objects. Key to our methods success is the use of initial shape estimation from a single camera view as a constraint in the grasp planner, coupled with the use of tactile sensing to adapt the grasp online. Despite training only in simulation, our approach works well in the real world and generates more precision grasps than the comparison approach [16].

REFERENCES

[1] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking In-Hand Manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 1 2020.

[2] Q. Lu and T. Hermans, "Modeling Grasp Type Improves Learning-Based Grasp Planning," *IEEE Robotics and Automation Letters*, 2019.

[3] Q. Lu, M. V. der Merwe, and T. Hermans, "Multi-Fingered Active Grasp Learning," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2020.

[4] A. Yamaguchi and C. G. Atkeson, "Combining Finger Vision and Optical Tactile Sensing: Reducing and Handling Errors While Cutting Vegetables," in *IEEE/RAS Intl. Conf. on Humanoid Robots*, 2016.

[5] F. R. Hogan, M. Bauza, O. Canal, E. Donlon, and A. Rodriguez, "Tactile regrasp: Grasp adjustments via simulated tactile transformations," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2018.

[6] B. Sundaralingam, A. S. Lambert, A. Handa, B. Boots, T. Hermans, S. Birchfield, N. Ratliff, and D. Fox, "Robust learning of tactile force estimation through robot interaction," in *IEEE Intl. Conf. on Robotics and Automation*, 2019.

[7] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?" in *Conference on Robot Learning*, 2017.

[8] A. Murali, Y. Li, D. Gandhi, and A. Gupta, "Learning to grasp without seeing," in *Intl. Symposium on Experimental Robotics*, 2018.

[9] F. Veiga, B. B. Edin, and J. Peters, "In-hand object stabilization by independent finger control," *Sensors*, vol. 20, no. 6, 2020.

[10] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Trans. on Robotics*, vol. 30, no. 2, 2014.

[11] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning Continuous 3D Reconstructions for Geometrically Aware Grasping," in *IEEE Intl. Conf. on Robotics and Automation*, 2020.

[12] J. Lundell, E. Corona, T. Nguyen Le, F. Verdoja, P. Weinzaepfel, G. Rogez, F. Moreno-Noguer, and V. Kyrki, "Multi-fingan: Generative coarse-to-fine sampling of multi-finger grasps," in *IEEE Intl. Conf. on Robotics and Automation*, 2021.

[13] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," in *Robotics: Science and Systems*, 2021.

[14] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," in *IEEE Intl. Conf. on Robotics and Automation*, 2013.

[15] Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active Tactile Object Exploration with Gaussian Processes," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2016.

[16] Q. Lu, M. V. der Merwe, B. Sundaralingam, and T. Hermans, "Multi-fingered grasp planning via inference in deep neural networks," *IEEE Robotics & Automation Magazine*, 2020.

[17] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *Intl. Journal of Robotics Research*, vol. 28, no. 7, 2009.

[18] M. A. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-d objects," *IEEE Trans. on Robotics*, vol. 25, no. 4, 2009.

[19] C. Rosales, L. Ros, J. M. Porta, and R. Suárez, "Synthesizing grasp configurations with specified contact regions," *Intl. Journal of Robotics Research*, vol. 30, no. 4, 2011.

[20] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *Intl. Journal of Robotics Research*, vol. 24, no. 4, 2005.

[21] D. Chen, V. Dietrich, Z. Liu, and G. von Wichert, "A probabilistic framework for uncertainty-aware high-accuracy precision grasping of unknown objects," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 1, 2018.

[22] K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard, and D. Kragic, "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," *IEEE Trans. on Robotics*, vol. 32, no. 4, 2016.

[23] M. S. Siddiqui, C. Coppola, G. Solak, and L. Jamone, "Grasp stability prediction for a dexterous robotic hand combining depth vision and haptic bayesian exploration," *Frontiers in Robotics and AI*, 2021.

[24] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann, "Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2006.

[25] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Simultaneous Localization and Grasping as a Belief Space Control Problem," in *Intl. Symposium on Robotics Research*, vol. 2, 2011.

[26] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *IEEE Intl. Conf. on Robotics and Automation*, 2015.

[27] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2015.

[28] A. Saxena, L. L. S. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information," in *AAAI National Conf. on Artificial Intelligence*, 2008.

[29] M. S. Kopicki, D. Belter, and J. L. Wyatt, "Learning better generative models for dexterous, single-view grasping of novel objects," *Intl. Journal of Robotics Research*, vol. 38, no. 10-11, 2019.

[30] H. Merzić, M. Bogdanović, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," in *IEEE Intl. Conf. on Robotics and Automation*, 2019.

[31] M. Veres, M. Moussa, and G. W. Taylor, "Modeling grasp motor imagery through deep conditional generative models," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, 2017.

[32] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2020.

[33] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *IEEE Intl. Conf. on Robotics and Automation*, 2021.

[34] B. Wu, I. Akinola, and P. K. Allen, "Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2019.

[35] J. Lundell, F. Verdoja, and V. Kyrki, "Ddgc: Generative deep dexterous grasping in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, 2021.

[36] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Intl. Conf. on Computer Vision*, 2019.

[37] H. Duan, P. Wang, Y. Huang, G. Xu, W. Wei, and X. Shen, "Robotics dexterous grasping: The methods based on point cloud and deep learning," *Frontiers in Neurorobotics*, vol. 15, 2021.

[38] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[39] B. Wu, I. Akinola, J. Varley, and P. K. Allen, "MAT: Multi-Fingered Adaptive Tactile Grasping via Deep Reinforcement Learning," in *Conference on Robot Learning*, 2019.

[40] H. Dang and P. K. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*, vol. 36, 2014.

[41] C. de Farias, N. Marturi, R. Stolkin, and Y. Bekiroglu, "Simultaneous tactile exploration and grasp refinement for unknown objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3349–3356, 2021.

[42] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[43] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *SIGGRAPH*, 2012.

[44] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," in *Robotics: Science and Systems*, 2012.

[45] N. Wettels, J. A. Fishel, and G. E. Loeb, "Multimodal tactile sensor," in *The Human Hand as an Inspiration for Robot Hand Development*. Springer, 2014, pp. 405–429.

[46] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, 1988.

[47] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *Intl. Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

[48] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163—169, Aug 1987.

[49] Dawson-Haggerty et al., "trimesh." [Online]. Available: https://trimsh.org/