Tactile Pose Feedback for Closed-loop Manipulation Tasks

Ota, Kei; Jain, Siddarth; Zhang, Mengchao; Jha, Devesh K.

TR2023-082 July 08, 2023

Abstract

Current generation manipulation systems operate in an open-loop fashion resulting in poor performance in the presence of disturbances. Robust manipulation requires a robot to compensate for uncertainties and errors arising due to contact in- teraction during manipulation. Consequently, it is essential that a robot can estimate an object's state and the relevant contact states so that manipulation can be controlled precisely. However, precise object state estimation is difficult due to occlusions and complex contact interactions during manipulation. This paper presents several different manipulation tasks where a robot may have to perform complex manipulation, which can introduce uncertainty leading to failure. To deal with this problem, we use in-hand pose estimation using vision-based tactile sensors to adjust our plan during manipulation. We present several different analyses for pose estimation using vision as well as tactile sensors to evaluate the importance of different modalities for these precision tasks. We demonstrate that using the proposed approach, we can perform the desired task successfully by incorporating feedback from the tactile pose estimation framework. See supplementary video at https://shorturl.at/eM125.

RSS 23 Workshop on Learning Dexterous Manipulation 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Tactile Pose Feedback for Closed-loop Manipulation Tasks

Kei Ota[1], Siddarth Jain[1], Mengchao Zhang[2] and Devesh K. Jha[1]

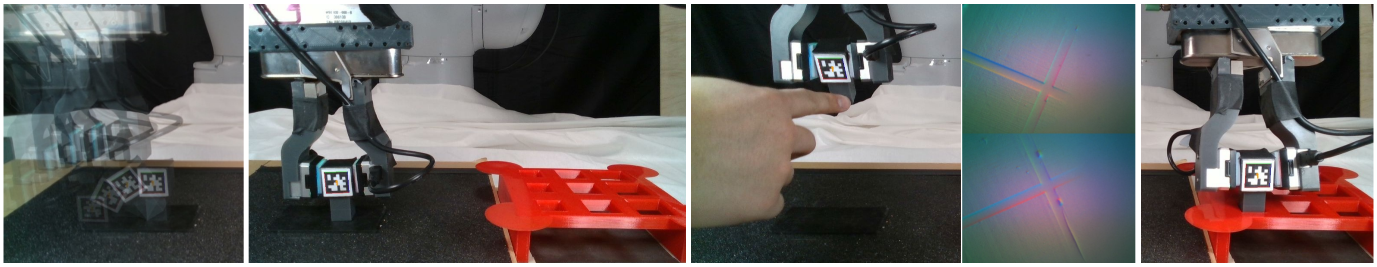
(a) Pivoting (b) Grasping (c) Pose estimation and correction (d) Insertion

Fig. 1: The experimental setups considered in this work. (a) A model-based planner pivots the peg using extrinsic contacts. (b) The robot grasps the peg using a rough pose obtained through AprilTag and a camera. (c) The peg is subjected to human-induced perturbation, wherein a human applies external force to the peg while in grsap of the robot. The robot estimates the in-hand pose using tactile images and a learned model. (d) The robot utilizes the estimated in-hand pose to correct the pose of the peg so that the robot can insert the peg into a hole with tight tolerances of approximately (0.5 mm).

***Abstract*—Current generation manipulation systems operate in an open-loop fashion resulting in poor performance in the presence of disturbances. Robust manipulation requires a robot to compensate for uncertainties and errors arising due to contact interaction during manipulation. Consequently, it is essential that a robot can estimate an object's state and the relevant contact states so that manipulation can be controlled precisely. However, precise object state estimation is difficult due to occlusions and complex contact interactions during manipulation. This paper presents several different manipulation tasks where a robot may have to perform complex manipulation, which can introduce uncertainty leading to failure. To deal with this problem, we use in-hand pose estimation using vision-based tactile sensors to adjust our plan during manipulation. We present several different analyses for pose estimation using vision as well as tactile sensors to evaluate the importance of different modalities for these precision tasks. We demonstrate that using the proposed approach, we can perform the desired task successfully by incorporating feedback from the tactile pose estimation framework. See supplementary video at https://shorturl.at/eM125.**

I. Introduction

Humans can perform very complex manipulation tasks in an effortless fashion due to their abilities to plan, sense, and react quickly. Consequently, average humans are very adept at performing a lot of dexterous manipulation tasks. It has been the long-standing goal of robotics to design systems that can perform robust manipulation and adjust for errors made during execution. However, such behavior is still elusive in most systems: they can only plan for simple manipulation, and more importantly, operate in an open-loop fashion. Designing closed-loop control methods for manipulation is essential for the successful deployment of manipulation systems. In this paper, we make use of in-hand pose estimation using vision-based tactile sensors to adjust to pose errors that could happen during multi-modal manipulation tasks, where the robot has to make and break contact with the object multiple times during execution.

Pose estimation is a fundamental problem in robotics [1], [2]. Despite a lot of work done in this field, it remains largely an open research problem and there is no one solution to it. Pose estimation is even more challenging for small-sized objects where more precision is required for tasks (consider, for example, the assembly of wire harnesses). Consequently, a lot of manipulation tasks still work in open-loop in heavily-structured environments as there is no principled method to perform closed-loop control of manipulation tasks. More recently, in-hand pose estimation using vision-based tactile sensors has been proposed for manipulation.

In this paper, we present the task of closed-loop peg insertion where a robot has to perform non-prehensile manipulation for grasping the peg before it can be inserted. The non-prehensile manipulation leads to errors in the localization of the peg due to inaccuracy in the vision system or unexpected slipping during manipulation execution. We show through multiple experiments that the open-loop plan seldom succeeds in insertion. We make use of in-hand pose estimation and localization using tactile sensors to estimate the object's state during grasping.

Note that in this work, we treat insertion as a placement task that will succeed with a precise pose estimate of parts. One can perform additional corrections using tactile and/or Force-Torque (F/T) sensors as proposed in [3], [4], [5] but this is not considered part of this study. We want to evaluate the accuracy of pose-estimation methods for these tasks without making use of additional pose-correction methods.

[1]K. Ota, S. Jain and D.K. Jha are with Mitsubishi Electric Research Labs, Cambridge, MA 02139 {ota,sjain,jha}@merl.com

[2]M. Zhang is with the Department of Mechanical Science and Engineering, UIUC, Urbana, IL, USA 61801. mz17@illinois.edu

II. Related Work

Pose estimation is a fundamental topic in robotics and thus there has been a lot of work in this area. While vision-based pose estimation of objects provides a rough estimate of the pose of objects in the environment of the robot, they are generally insufficient for a lot of precise manipulation tasks. For example, consider the task of robotic insertion where the robot has to insert a peg into a hole. For such tasks, localization of parts needs to be within the tolerance of assembly (which is generally around a few millimeters or sub-millimeters). Such precision is difficult to achieve using vision alone [4], [3]. Consequently, vision-based tactile sensors are getting a lot of attention in recent research for perception tasks.

Vision-based tactile sensors have been very popular recently for the high-resolution perception capabilities these sensors offer. They have been used for various different tasks including insertion [3], [6], feedback control [7], [8], extrinsic contact state estimation [9], [6], grasp stabilization [10], [11]. More recently, they have been also used for pose estimation and localization of objects. An illustrative example is the recent work proposed by Bauza et al. [12], which proposes a novel algorithm capable of predicting object type and pose by leveraging geometric models of diverse objects. Specifically, upon contact formation, the tactile sensors generate high-detailed images of the contact patch, which the proposed algorithm utilizes to identify the most probable object pose. Similarly, in [13], authors present an interactive perception technique for precise localization and identification in a multi-object assembly scenario using Gelsight sensors.

Tactile sensors have previously been used for designing various kinds of closed-loop control systems. In [3], a Reinforcement Learning (RL)-based insertion policy was proposed which can perform insertion for objects with different geometries. In [7], [8], feedback from tactile sensors was used for stabilization and tracking of manipulation trajectories during different manipulation tasks. In [9], the estimation from tactile sensors could be used for the stable placement of an object while minimizing frictional forces during the resulting interaction.

In this paper, we study the use of in-hand pose estimation using tactile sensors for closed-loop control of multi-modal manipulation tasks where errors can accumulate due to long-horizon nature of the tasks and various contact interactions.

III. Problem Statement

In this section, we present a formal statement of the problem we study in this paper along with the related assumptions. We consider the task of peg insertion, involving manipulation of the pose of the peg prior to insertion, and disturbance by a human after grasping the peg. A planar description of the problem is shown in Fig. 2, where the robot needs to perform non-prehensile manipulation for the initial transition (shown as $a \rightarrow b$). Visualization of the whole pipeline is shown in Fig. 1. We make the following assumptions in this problem:

1) The geometry of the peg is perfectly known.
2) The frictional parameters of the different contact interactions are perfectly known.
3) All objects are rigid.
4) The hole location is perfectly known to the robot.

We would like to discuss the implications of the above assumptions briefly. Assumptions 1, 2, and 3 are very common in model-based manipulation planning. Assumption 4 could be relaxed for insertion problems. This can be replaced by the design of a suitable hole detection algorithm that could be precise in the detection of holes [5], [14]). These methods generally lead to some imprecision during insertion which can be compensated using some methods using active exploration [3], [5]. However, we do not consider such a feedback mechanism here. Under the above assumptions, we try to make use of in-hand pose estimation using tactile sensors to estimate the error accrued during the manipulation $a \rightarrow b$.

IV. Method

In this section, we present the approach for designing the closed-loop manipulation system. In particular, we explain a model-based manipulation that takes a point cloud description of the object geometry and physical parameters of the environment and computes a feasible manipulation trajectory for non-prehensile manipulation. Then we present a learning-based method that can be used for estimating the pose of the grasped peg using tactile images.

A. Manipulation Planning

We make use of a novel contact-implicit trajectory optimization (CITO) for planning the proposed manipulation [15], [16], [17], [7]. We embed the problem of contact selection during trajectory optimization as part of the optimization [18], [19]. Details of the method are presented in the paper [18], [19] and we briefly explain it here. The constraints that need to be satisfied for each point on a trajectory for the particular task are the following:

1. Bound Constraint:

$$q_t \in \mathcal{Q},\ u_t \in \mathcal{U},\ z_t(y) \in \mathcal{Z}\ \forall y \in Y \tag{1}$$

2. Distance Complementarity Constraint:

$$0 \leq z_t(y) \perp g(q_t, y) \geq 0 \quad \forall y \in Y \tag{2}$$

3. Constraint on $z_t(y)$:

$$h(q_t, y, z_t(y)) \geq 0 \quad \forall y \in Y \tag{3}$$

4. Constraint on Control:

$$c(q_t, u_t) \geq 0 \tag{4}$$

5. Integral Constraint:

$$\underbrace{s_{q,u}(q_t, u_t) + \int_{y \in Y} s_z(q_t, y, z_t(y))dy}_{=:s(q_t, u_t, z_t; Y)} = 0 \tag{5}$$

Then the full optimization problem can be written by enforcing these constraints at all the points along the trajectory using the mechanics of the problem in Fig. 2. We formulate the following infinite programming with complementarity constraints trajectory optimization problem denoted as $P(Y)$:

$$\min_{q, \dot{q}, u, z} f(q, \dot{q}, u, z) \tag{6a}$$

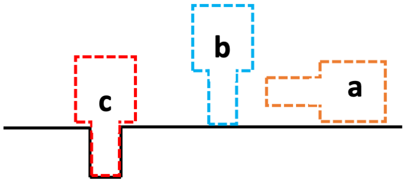

Fig. 2: This paper considers designing a precise controller for manipulating the peg from an initial state $a \to b \to c$. The manipulation from $a \to b$ leads to some error due to the imprecision of the models and physical parameters, which is compensated in $b \to c$ using pose estimation using vision and tactile.

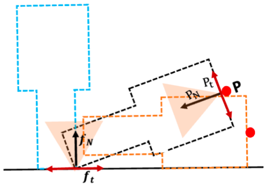

Fig. 3: The pivoting manipulation problem in SE(2) that we study for the non-prehensile manipulation in this paper.

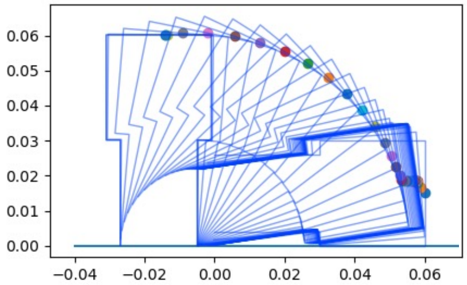

Fig. 4: A feasible trajectory computed by the CITO planner. The colored circles denote the location of the manipulator's finger. As could be seen, the manipulator maintains slipping contact with the face of the peg during manipulation. The CITO computes a feasible position as well as force trajectory for the pivoting manipulation.

$$\text{s.t. } q_0 = q_{start}, q_T = q_{goal} \tag{6b}$$

$$\dot{q}_t \in \dot{\mathcal{Q}}\ \forall t \in \mathcal{T} \tag{6c}$$

$$q_t - q_{t+1} + dt\dot{q}_t = 0 \quad \forall t \in \mathcal{T} \tag{6d}$$

$$\begin{gathered} 0 \le v(q_t, \dot{q}_t, y) \perp h(q_t, y, z_t(y)) \ge 0 \\ \forall y \in Y,\ \forall t \in \mathcal{T} \end{gathered} \tag{6e}$$

where $f(q, \dot{q}, u, z) := \sum_{t\in\mathcal{T}}[f_{q,\dot{q},u}(q_t, \dot{q}_t, u_t) + \int_{y\in Y} f_z(q_t, y, z_t(y))dy]$, dt is the time step duration, and $\mathcal{T} = \{0, \ldots, T-1\}$ with T the total number of time steps in the trajectory. For the sake of brevity, we use the notation $q = [q_0, \cdots, q_T]$, $\dot{q} = [\dot{q}_0, \cdots, \dot{q}_{T-1}]$, $u = [u_0, \cdots, u_{T-1}]$, $z = [z_0, \cdots, z_{T-1}]$. With a little abuse of notation, we use $z_t = [z_t(y)\ \forall y \in Y]$ where $z_t(\cdot)$ is the mapping and z_t is a concatenation of all the instantiated variable for all $y \in Y$.

B. In-hand pose estimation and correction

We train a model to estimate an in-hand pose of an object to correct its pose error when grasping the object prior to insertion. The model takes two tactile images from tactile sensors attached to the two fingers $I^{\text{left}}, I^{\text{right}}$, and estimates the pose error in robot frame $(\hat{\mathrm{d}x}, \hat{\mathrm{d}z}, \hat{\mathrm{d}\theta})$. We collect data in the real system with random SE(2) displacements $(\mathrm{d}x, \mathrm{d}z, \mathrm{d}\theta)$ from a calibrated center pose to train the model. After estimating the displacements, the robot corrects the pose by computing the displacements in the local frame of the manipulator.

V. Experiments

In this section, we present different experiments which are conducted to answer and explain the following questions:

1) What is the accuracy of the different pose estimation methods described in the paper?
2) What is the success rate for the planning algorithm in the execution of the proposed task?
3) What is the success rate of the different closed-loop control frameworks and what degree of precision could be obtained by the different methods?

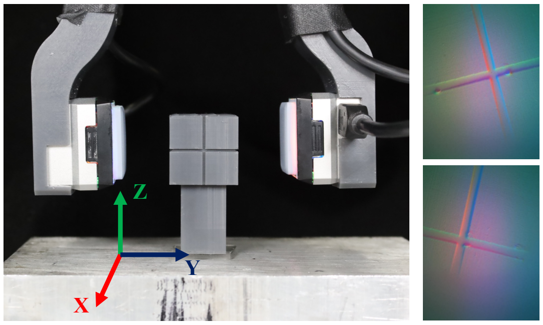

Fig. 5: Data collection setup. **Left**: Grippers with GelSight Mini tactile sensors are used for data collection. A peg on the center has grooves with cross textures on all surfaces from which a trainable model can estimate displacement from the center position. **Right**: Example tactile images.

The above questions help us understand the strengths and weaknesses of the different modules which can be used to design a closed-loop system for performing high-precision manipulation tasks. In the following sections, we answer the above questions through hardware experiments.

A. Experimental Settings

Tactile sensor. We use a commercially available GelSight Mini [20] tactile sensor, which provides 320×240 compressed RGB images at a rate of approximately 25 Hz, with a field of view of 18.6×14.3 millimeters.

Robot platform. The MELFA RV-5AS-D Assista robot, a collaborative robot with 6 DoF, is used in this study. The tactile sensor is mounted on the WSG-32 gripper (see Fig. 5). We use a Force-Torque (F/T) sensor, which is mounted on the wrist of the robot. The F/T sensor is used to implement an indirect force controller for implementing the force trajectory obtained from the manipulation planner. This force controller uses the default stiffness controller of the position-controlled robot.

Design of pegs. In order to estimate the pose of the target peg inside the gripper, a cross-shaped groove was created at

TABLE I: Average errors in estimating displacements in SE(2) from the center for the three pegs with different sizes. The results demonstrate the regression approach achieves better performance, specifically it achieves submillimeter errors in estimating displacements in translation. The numbers in bold characters denote the best results. It is noted that the images in *M* are sampled from the same distribution with the training set.

Size		Regression	Contrastive
M	$\mathrm{d}x$	$\mathbf{0.01}$	0.46
	$\mathrm{d}z$	$\mathbf{0.01}$	0.27
	$\mathrm{d}\theta$	$\mathbf{0.05}$	1.38
S	$\mathrm{d}x$	$\mathbf{0.32}$	3.21
	$\mathrm{d}z$	$\mathbf{0.15}$	2.65
	$\mathrm{d}\theta$	$\mathbf{0.73}$	12.00
L	$\mathrm{d}x$	$\mathbf{0.10}$	0.73
	$\mathrm{d}z$	$\mathbf{0.03}$	2.24
	$\mathrm{d}\theta$	$\mathbf{0.40}$	2.87

TABLE II: Task completion comparison over 10 trials for the three different peg sizes and tolerances, and two different settings, *Vertical* and *Tilted*. In the *Vertical* setting, the robot pivots the peg and then tries to insert it in the hole. In the *Tilted* setting, we add disturbance by pushing or pulling the peg with a human hand to see the robustness (see Fig. 1). Regarding the method, *Vis* indicates the results with only a vision-based model, where the pose of the peg is given by the AprilTag. The *Vis + Tac* corrects the in-hand pose of the object by utilizing the tactile images and the trained model. The numbers in bold characters denote the best results.

		Vertical		Tilted		
Peg size	Tolerance	Vis	Vis + Tac	Vis	Vis + Tac	Regrasp
S	0.5 mm	0/10	**10/10**	0/10	**8/10**	**10/10**
	1.0 mm	2/10	**10/10**	0/10	**9/10**	**10/10**
	2.0 mm	4/10	**10/10**	0/10	**10/10**	**10/10**
M	0.5 mm	0/10	**10/10**	0/10	**9/10**	**10/10**
	1.0 mm	3/10	**10/10**	0/10	**10/10**	**10/10**
	2.0 mm	5/10	**10/10**	0/10	**10/10**	**10/10**
L	0.5 mm	0/10	**9/10**	0/10	**7/10**	**10/10**
	1.0 mm	2/10	**10/10**	0/10	**8/10**	**10/10**
	2.0 mm	3/10	**10/10**	0/10	**10/10**	**10/10**

the center of all surfaces of the peg as shown in Fig. 5. This is based on the realization that the tactile sensors need some features on the observed contact to localize. In the absence of any such features, a perception task would most likely fail. Three pegs of varying square sizes, namely $\{17, 22, 27\}$ mm, were fabricated via 3D printing, and we denote them as {*S*, *M*, *L*}. Each peg is equipped with a head part that was 8 mm larger in size. Additionally, corresponding holes are 3D-printed with clearance sizes of $\{0.5, 1.0, 2.0\}$ mm for each peg to assess the efficacy of insertion performance under conditions of tight clearances.

B. Pose estimation

First, to evaluate the in-hand pose estimation performance on different computer vision methods, we measure the pose estimation errors on different two loss functions: regression (MSE) and contrastive loss [21], [22] with the same backbone network, Vision-Transformer [23].

Settings To train the models, a dataset of 8K images was collected from the real system, specifically using a medium-sized (17 [mm]) peg. To generate a diverse range of images with various possible displacements, we introduced random displacements in SE(2), where $\mathrm{d}z$ and $\mathrm{d}x$ were drawn from a uniform distribution $U(-5, 5)$ [mm], and $\mathrm{d}\theta$ was drawn from $U(-45, 45)$ [deg]. These displacements were added from the center position of the peg, as illustrated in Fig. 5. To assess the effectiveness of the trained models, an additional set of 1K images was collected for each peg, and another set of 1K images is used for validation. These images were collected from the same uniform distribution as the training set.

Results Table I represents the average errors in predicting the displacements in SE(2). The regression approach outperforms the contrastive learning approach. Common failure case of contrastive learning is shown in Fig. 6 in Appendix.

C. Peg insertion

Settings Next, we utilize the trained pose estimation model for peg insertion tasks in three different settings. *Vertical* setting considers peg insertion without any external perturbation, resulting in a small error in the orientation of the peg ($\mathrm{d}\theta$) as we consider the table flat. *Tilted* setting considers the peg insertion with human perturbation as shown in Fig. 1(c), an external force by a human hand is applied to the peg after grasping, resulting in requiring more pose correction, especially for the orientation θ, compared to the vertical insertion. Finally, we also test on *Regrasping* setting, where we consider the same disturbance as *Tilted* setting, but the robot re-grasps the peg by placing the peg onto the table and picks it again. This enables the peg orientation $\mathrm{d}\theta$ to be close to 0, resulting in easier insertion.

Results Table II shows the task completion numbers over 10 trials. The vision-based model (Vis) fails to complete the task because the accuracy of the AprilTag is not enough for tight insertion on both *Vertical* and *Tilted* settings. Our method, which corrects pose errors after grasping, boosts the performance to close to 100% even on the *Tilted* setting, however, it still fails to insert especially for the large peg, where small errors in angle have bigger effects when insertion. We demonstrate that re-grasping the peg after placing the onto the flat surface simplifies the task and improves the performance to nearly 100% accuracy. See supplementary video at https://shorturl.at/eM125.

VI. Conclusions

Most manipulation systems operate in open-loop where a robot can not observe and react to changes in system states from the planned trajectory. In this paper, we study a closed-loop control framework for a high-precision assembly task where a robot has to perform non-prehensile manipulation before grasping a part to be assembled. We proposed in-hand pose estimation using vision-based tactile sensors for performing closed-loop manipulation during a multi-modal manipulation task.

References

[1] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3343–3352.

[2] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 2020, pp. 574–591.

[3] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-RL for insertion: Generalization to objects of unknown geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6437–6443.

[4] D. K. Jha, D. Romeres, W. Yerazunis, and D. Nikovski, "Imitation and supervised learning of compliance for robotic assembly," in *2022 European Control Conference (ECC)*, 2022, pp. 1882–1889.

[5] D. K. Jha, D. Romeres, S. Jain, W. Yerazunis, and D. Nikovski, "Design of adaptive compliance controllers for safe robotic assembly," *arXiv preprint arXiv:2204.10447*, 2022.

[6] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 241–10 247.

[7] Y. Shirai, D. K. Jha, A. U. Raghunathan, and D. Hong, "Tactile tool manipulation," *arXiv preprint arXiv:2301.06698*, 2023.

[8] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez, "Tactile dexterity: Manipulation primitives with tactile feedback," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 8863–8869.

[9] S. Kim, D. K. Jha, D. Romeres, P. Patre, and A. Rodriguez, "Simultaneous tactile estimation and control of extrinsic contact," *arXiv preprint arXiv:2303.03385*, 2023.

[10] S. Dong, W. Yuan, and E. H. Adelson, "Improved gelsight tactile sensor for measuring geometry and slip," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 137–144.

[11] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining grasps within slipping bounds by monitoring incipient slip," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3818–3824.

[12] M. Bauza, A. Bronars, and A. Rodriguez, "Tac2pose: Tactile object pose estimation from the first touch," *arXiv preprint arXiv:2204.11701*, 2022.

[13] K. Ota, D. K. Jha, H.-Y. Tung, and J. B. Tenenbaum, "Tactile-filter: Interactive tactile perception for part mating," *arXiv preprint arXiv:2303.06034*, 2023.

[14] D. K. Jha, S. Jain, D. Romeres, W. Yerazunis, and D. Nikovski, "Generalizable human-robot collaborative assembly using imitation learning and force control," *arXiv preprint arXiv:2212.01434*, 2022.

[15] A. U. Raghunathan, D. K. Jha, and D. Romeres, "Pyrobocop: Python-based robotic control & optimization package for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 985–991.

[16] Y. Shirai, D. K. Jha, A. U. Raghunathan, and D. Romeres, "Robust pivoting: Exploiting frictional stability using bilevel optimization," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 992–998.

[17] A. U. Raghunathan, D. K. Jha, and D. Romeres, "Pyrobocop: Python-based robotic control & optimization package for manipulation and collision avoidance," *arXiv preprint arXiv:2106.03220*, 2021.

[18] M. Zhang, D. K. Jha, A. Raghunathan, and K. Hauser, "Stocs: Simultaneous trajectory optimization and contact selection for contact-rich manipulation," in *Embracing Contacts-Workshop at ICRA 2023*, 2023.

[19] M. Zhang, D. K. Jha, A. U. Raghunathan, and K. Hauser, "Simultaneous trajectory optimization and contact selection for multi-modal manipulation planning," 2023.

[20] "GelSight Mini," https://www.gelsight.com/gelsightmini/, accessed: 2023-01-16.

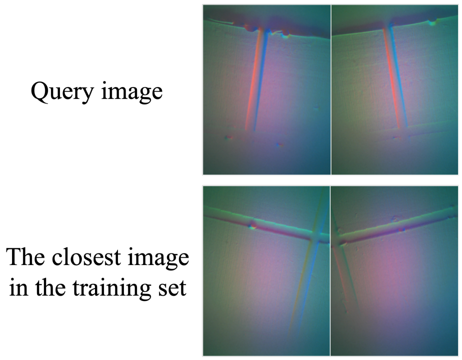

Fig. 6: An example of failure case of contrastive learning, which can be attributed to its susceptibility to outliers. This can happen due to unanticipated image features not being captured in the training dataset. Specifically, the loss function used in contrastive learning aims to identify the most similar image in the training set given an input image. Consequently, image features not included in the training set, such as the edges or corners of a small peg in our study, may not be recognized by the model.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9726–9735.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

Appendix

A. *Failure cases in contrastive learning*

The contrastive learning model has poorer performance in estimating the pose of the peg. Specifically, it seeks to identify the most similar image within the training set, which renders it susceptible to outliers such as edges of corners of pegs in the captured images that are not present in the training set, even if the relative pose is identical. We show an example of this in Fig. 6.