

Fingers See Things Differently (FIST-D): An Object Aware Visualization and Manipulation Framework Based on Tactile Observations

Chenxi Xiao^{1b}, Naveen Madapana^{1b}, and Juan Wachs^{1b}

Abstract—Planning object manipulation policies based on tactile observations alone is a challenging task due to the multi-factorial variances in the measured point cloud (e.g. sparsity, missing regions, rotation, etc.) and the limited sensory information available through tactile sensing. Nevertheless, the mainstream grasp planners are designed for well-structured point cloud data, and lack the crucial ability to plan grasps in unexplored regions that are common during tactile sampling. Hence, it is crucial to detect the grasp regions from incomplete and unstructured tactile point cloud data. To address this limitation, we propose a novel framework that utilizes a support set of CAD models to augment the tactile observations, and thereby facilitate object recognition, visualization, and developing manipulation policies solely using tactile samples. To cope with the noise and sparsity of tactile observations, we propose uGPIS, a surface reconstruction method that utilizes the occupancy possibility function and the Gaussian Process Regression to recover the underlying surface from tactile point clouds. Then, we complete the partially observed tactile point cloud using the prior knowledge obtained from the support set of full CAD models. This prior information will provide the enriched geometric information that is crucial to determine the grasp regions. Our experimental results on a physical simulation show that our method can successfully combine the prior knowledge from the database to enhance the grasp success rate.

Index Terms—Deep learning in grasping and manipulation, force and tactile sensing, perception for grasping and manipulation.

I. INTRODUCTION

TACTILE sensing can be an essential requirement in a variety of robotic applications, especially when visual information is unavailable (e.g. occlusion, smoke and darkness, reflective and transparent objects, etc.) [49]. For example, robots with tactile sensing are deployed to explore objects buried under the sand or seabed, or objects purposefully concealed to reduce the exposure to risks for field technicians. Similarly, applications in our daily lives and industrial fields are vast and disparate,

Manuscript received October 15, 2020; accepted February 12, 2021. Date of publication March 8, 2021; date of current version April 9, 2021. This letter was recommended for publication by Associate Editor Y. Yang and Editor M. Vincze upon evaluation of the Reviewers' comments. This work was supported by National Science Foundation under Grant NSF NRI #1925194. (Corresponding author: Juan Wachs.)

The authors are with Intelligent Systems and Assistive Technologies Lab at Purdue University, West Lafayette, Indiana 47907 USA (e-mail: xiao237@purdue.edu; nmadapan@purdue.edu; jpwachs@purdue.edu).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2021.3064211>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3064211

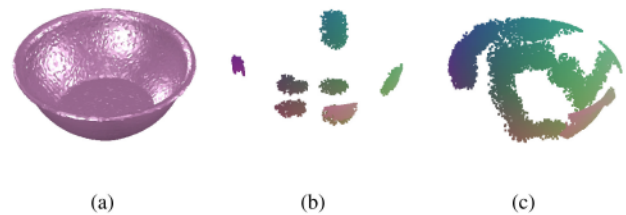


Fig. 1. Tactile point clouds have complex point distributions that are intractable to be modeled. (a) Ground-truth bowl CAD model, (b) Point cloud sampled by discrete contacts, and (c) Continuous contacts.

such as manipulating objects inside paper or vinyl wrappers, exploring and recognizing warehouse objects wrapped in shock-absorbing fabric covers, inferring the shape of human bodies dressed in clothes [21].

While humans can manipulate objects and comprehend their properties purely based on tactile feedback, it is a challenging task for robotic systems to achieve the same due to inherent perceptual differences between robots and humans. One important difference lies in how the prior knowledge about objects is used. In contrast to robots that rely mostly on vision [15], [16], [19], [40], human decision making is based on holistic sensory information including visual, auditory and tactile modalities. Furthermore, humans tend to effectively utilize abundant prior knowledge about versatile objects acquired through life-long learning to assess, plan and manipulate new objects [12]. In contrast, most robots are trained on a much smaller set of objects and they do not generalize sufficiently well to new or unseen objects [7], [31].

Nevertheless, the local and discrete features acquired by machine tactile feedback are not as indicative of the object's overall properties as its visual counterparts. For example, Fig. 1(a) is a more informative observation compared to the incomplete counterparts depicted in the Fig. 1(b) and (c). This is evident from the fact that tactile-based intelligent systems generally perform poorly even though a learning system has been involved [24]. Furthermore, in a teleoperation system, the insufficient transparency of tactile feedback also degrades the teleoperator's decision-making performance due to the increased cognitive load [30]. In this regard, we believe that a key question that can push the limits of tactile reasoning is the following: how can a robot use the loosely coupled tactile samples to infer the

critical object-related properties (e.g. geometry, center of mass, etc.) that can be used for decision making and motion planning?

We identified through experiments (refer to Section IV) that making inferences based on a tactile point cloud is a difficult task due to inductive bias produced by the following variances. First, the distribution of tactile samples is a result of both the information acquisition policies and the sensing modalities. Further, this distribution alters greatly due to the inherent differences between individuals (or autonomous agents), methods, decision variables and even random factors [51]. The inductive bias is further aggravated by several variances including point sparsity [50], unknown object poses [45], positional offset, observed contact regions and also the novelty of the object (a new object not seen before by the user in the same context).

In this letter, we propose FIST-D, the first known comprehensive framework for visualizing, recognizing and planning manipulation strategies based purely on tactile observations. In contrast to previous works, our framework is robust to inductive bias and can effectively plan grasps based on the objects inferred from a support set. We further extend the framework's ability to generalize to unknown objects by proposing uGPIS algorithm, a novel surface estimator that utilizes partially observed tactile point clouds to facilitate both contact point visualization and grasp planning. The main contributions of this work are as follows:

- We propose FIST-D, a framework for planning manipulation strategy based on tactile observations.
- We propose a method that is robust towards inductive bias for identifying similar objects from a support set.
- We introduce the uGPIS algorithm, a surface estimator that can reconstruct non-watertight surfaces based on occupancy possibility.
- We implement a grasp generator for a two-finger non-parallel jaw gripper, and evaluate the algorithm on a simulated Taurus robot gripper.
- We develop an interactive simulation environment for tactile sampling.

II. RELATED WORK

A. Robot-Assisted Tactile Sampling & Exploration

Tactile sensing has been popularly used to explore environments when vision is either not applicable (e.g. dimmed areas, underwater, or when objects are concealed) or not accurate. For this reason, previous works have focused on exploring the workspace by either visuo-tactile integration [38], [42], or purely using tactile sensing [11], [24]. The latter case can be categorized into four sub-categories by the way the tactile exploration is performed: 1) Exploration based on pre-defined motion [24], 2) Exploration over regions with high positional uncertainty [6], [11], [35], 3) Exploration through minimization of the action cost by efficient sampling [50], and 4) Human-guided exploration through teleoperation [39]. In our approach, a human operator guides or teleoperates the remotely located robot to explore the surface, and thereby, our model falls into the fourth category.

B. Tactile-Based Shape Estimation

In general, information sampled by tactile sensors consists of discrete points that only contain local information [17]. Therefore, inferring the underlying geometries behind multiple contact points is a crucial step towards developing reasoning ability and creating visualization to facilitate tactile based human-robot interaction [46]. To infer the geometric structures from contacts, Charlebois *et al.* [4] obtained the local surface by fitting and combining B-spline patches. Meier *et al.* [20] performed tactile shape reconstruction by using the Kalman filter. More recently, Ottenhaus *et al.* and Rosales *et al.* showed that Gaussian Process Implicit Surface [44] is an attractive method to pursue due to the smoothness of the regressed surface as well as the robustness towards sparsity [29], [35]. However, all previous approaches still rely on the Signed Distant Field (SDF) function, which is not directly accessible and can only be acquired by estimation [29]. Nevertheless, estimating the SDF robustly is a challenging task due to the noisy surface normal measurements, inconsistent local curvatures, presence of thin structures, etc. As an improvement, we propose uGPIS algorithm that reconstructs objects by utilizing the occupancy possibility function rather than the estimated SDF samples.

C. Grasp Detection

Grasping is a primitive action for robots to interact with objects. As mentioned in the survey conducted by Shimoga *et al.* [37], grasp detection is a persistent challenge in object manipulation. The grasp detection task requires an algorithm to generate the manipulator's target state conditioned on the object's observation. Among works that utilized two-finger grippers, a majority of them focus on grasp detection on parallel jaws [16], [40]. In comparison, we designed a grasp detection algorithm and the corresponding evaluation metric for *non-parallel jaws*.

Grasps are generally being planned using either of the two approaches: *model-based grasping* and *model-free grasping*. *Model-based grasping* [13]–[15] often relies on a set of “template” objects (i.e. support set) with pre-planned grasps. During testing, the pre-planned grasps are retrieved and aligned with the observation from the sensor. However, this approach is hindered by any type of mismatches in alignment (e.g. shape discrepancy, alignment error in object pose, etc). For this reason, it is not feasible to plan grasps for novel objects. To tackle this issue, *model-free grasping* has been developed, which plans grasps without database priors. In this paradigm, grasps are either generated by searching [16], [40] or deep generative models [23].

While the *model-free* method has reportedly achieved *state-of-the-art* performance with visual observations, it is intractable to be applied directly to the point clouds of low quality (e.g. incomplete, sparse, and noisy, as is often the case with tactile points) [36], [40], [41]. To detect grasps reliably on tactile points, we use a mixed paradigm for grasp detection. First, we leverage a *model-based* method to detect grasps on models that match the tactile observation. We use point cloud registration metric as an indicator of whether the *model-based* grasp detection is successful. When grasp detection is unsuccessful, we rely on

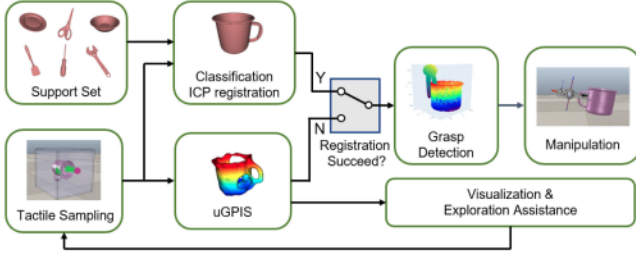


Fig. 2. The flowchart of our proposed framework. Our approach can comprehend the object geometries in two conditions: 1) Matching with the object models from the support set. If successful, the CAD templates can provide superior performance in visualization and grasp detection; 2) If template objects are not available in the support set, we leverage uGPIS to reconstruct the underlying surface of tactile observations.

TABLE I
COMPARISON OF CLASSIFICATION ACCURACY (IN %) ON THE TACTILE POINT CLOUD DATA FROM INTERACTIVE TACTILE SAMPLING

Method	PointNet[33]	DGCNN[44]	Triangle-Net[46]
Top-1 Acc (w/o aug)	12.3	8.2	43.8
Top-5 Acc (w/o aug)	37.0	30.1	67.1
Top-1 Acc (aug)	19.2	15.1	47.9
Top-5 Acc (aug)	38.4	39.7	76.7

model-free grasp planner to detect grasps on high-resolution surfaces reconstructed by uGPIS algorithm.

III. METHODS

Our framework is focused on the geometric information inferred from contacts. The proposed approach can be summarized in the flowchart shown in Fig. 2. After receiving the tactile point cloud as an input, we leverage on uGPIS algorithm (refer to the Section III-B) to create an implicit surface representation of the object with infinitely high resolution. Using such an augmented surface representation is beneficial in at least two aspects: 1) The sparse and noisy points can be visualized with enhanced quality, and 2) It allows the grasp detection algorithm to use dense point clouds rather than the sparse tactile point clouds.

Next, we utilize the support set to complete the partially observed tactile point cloud. We attribute the partial point cloud to the most similar CAD models in the support set, and then align the partial observation with CAD models by point cloud registration (refer to Section III-A). If the registration is successful, then we perform grasp detection on the full CAD model, which helps to detect grasps on regions that have not been explored. When the registration fails (e.g. large registration error for an object from an unseen category), we detect grasps using the extracted implicit surface obtained from the uGPIS.

A. Object-Aware Shape Inference

In order to expand the region where grasp detection can be applied, we infer the completed shape from partially observed point cloud data. However, the variances discussed in the Section I can result in severe inductive bias that has apparent negative effects for many deep learning approaches (refer to Table I). To acquire robustness to the inductive bias, we propose a two-stage shape

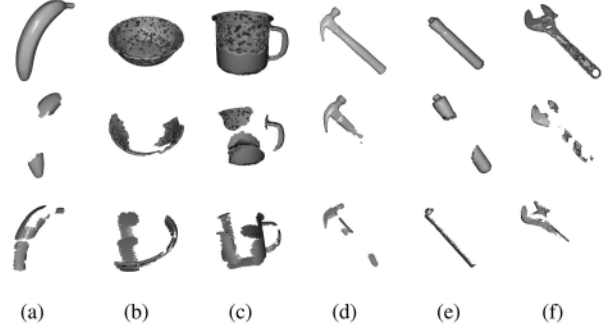


Fig. 3. Qualitative comparison of full CAD objects (row 1), generated partial observations by Alg. 1 (row 2), and tactile observations (row 3).

inference method. First, we use a robust point cloud classifier to propose a candidate object set $\{X_1, X_2, \dots, X_k\}$, with each object being represented as a point cloud. Next, we align the tactile observation X^p with each candidate object by rigid point cloud registration.

Since we align X^p with multiple candidate objects, we find that ICP loss (Eq. 1) can effectively distinguish the correctly aligned object from the others. A match is successful if an object has the minimum loss among all candidate objects (loss is defined in Eq. 1). Otherwise, the alignment failure is recognized and the object model is represented by the implicit surface obtained by the uGPIS algorithm.

Object Recognition: Human-guided exploration involves proactive sampling which may generate tactile samples with complex distributions that are different from either full CAD models or visual perceptions (refer to Fig. 1). Though the object's CAD models are available, an effective deep learning method that utilizes these models does not exist due to the inductive bias resulting from the human-guided exploration patterns. We show that the inductive bias can be mitigated by training a network using the algorithm shown in Alg. 1. For results, please refer to our quantitative evaluation in Table I. This algorithm takes in a set of anchor points located on the surface of a CAD object model M . Next, we perform “Expand” operation (Alg. 1 line 2) which collects the vertices in the CAD model M that are close to the anchor points in breadth-first order sequence. The collection stops when the number of collected points has reached r_t percentile of all vertices in M . Then, we perform “Erosion” operation on the point set P_s acquired from “Expand”. In “Erosion,” we remove r_s percentile of the total points from P_s that are close to the anchor points P_A^2 in a breadth-first order sequence. We qualitatively demonstrate the visual effect of generated meshes in the Fig. 3.

Registration: Next, the obtained tactile observations are aligned with the candidate CAD models via registration. Several methods have been used in the literature to accomplish registration, including ICP [1], CPD [26], FilterReg [8], SVR [3], etc. Since the tactile point cloud is sparse and incomplete, it can potentially result in noisy and non-expressive features that are not suitable for matching *point correspondences* [10]. To resolve this issue, the alignment is reinitialized and repeated with a large number of different initial poses. The best alignment is chosen

Algorithm 1: Generate Training Data.

Input: M : A CAD mesh model
 P_A : A set of anchor points in M
 r_t : Ratio of Expand operation
 r_s : Ratio of Erosion operation
Result: P_R : Object's partially observed point cloud

```

1 Algorithm generate_ruptured_mesh
2    $P_s = \text{Expand}(M, P_A, r_t)$ 
3    $P_A^2 \leftarrow$  random select  $k$  anchor points from  $P_s$ 
4    $P_R = P_R \cup \text{Erosion}(P_s, P_A^2, r_s)$ 
5   return  $P_R$ 

```

according to the following loss function (1):

$$\mathcal{L}_{ICP}(\mathbf{T}) = \frac{1}{n_q} \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{X}} \|\mathbf{p} - \mathbf{T}\mathbf{q}\|^2 \quad (1)$$

Where $\mathbf{p}, \mathbf{q}$ are the source and target point clouds respectively, n_q is the number of points in $\mathbf{q}$, $\mathcal{X}$ is the correspondence relationship between $\mathbf{p}$ and $\mathbf{q}$ (e.g. nearest neighbourhood), and $\mathbf{T}$ is the transform matrix. Since the matrix $\mathbf{T}$ obtained from SVR [3] already includes the scaling factor, our approach remains robust to the variations in the object scale.

B. Implicit Surface Reconstruction

Gaussian Process Implicit Surface (GPIS) [44] is an approach that has been widely adopted to estimate the underlying continuous manifold from the noisy and sparse tactile point clouds [19], [35]. The working principle of GPIS can be explained in two steps: 1) Fit an implicit function of the Signed Distant Field (SDF) $s = f(x)$ by the Gaussian Process Regression [33], and then, 2) Extract the 0-isosurface of the function f by Marching Cubes [28].

Nevertheless, we notice that GPIS has a few drawbacks that limit its performance in terms of robustness and representation ability. First, GPIS is only applicable to watertight surfaces, but many objects cannot be identified as watertight objects when the thickness cannot be measured well (e.g. a piece of paper, thin cable, etc.), especially when the spatial resolution of many tactile sensors is not sufficiently high. For example, a cable wire may be observed as a one-dimensional point array with zero thickness. In that case, the watertight property may no longer hold true, which would result in a failed reconstruction by GPIS approach. Second, GPIS requires SDF samples from both sides of the surface, but the points beneath the object are usually inaccessible by touches [29]. Previous works tackled this issue by generating SDF anchor points beneath the object surface by shifting the detected contact points along the contact normal vectors at a distance d_n [19], [35]. However, choosing the value of d_n requires careful parameter tuning. Otherwise, it may lead to either convergence issues during regression, or undesirable reconstruction results (e.g. inaccurate surface curvatures, refer to the experiment in Section IV-C). For this reason, the expressive ability of GPIS is limited for tactile observations.

In this regard, we propose unilateral GPIS (uGPIS), a robust surface reconstruction method based on the estimation of the occupancy possibility function $f_o: \mathbb{R}^1 \in [0, 1]$, indicating the possibility that a position is occupied. The use of the occupancy possibility function can potentially address the previous concerns as it allows objects to be non-watertight, and it does not discriminate between inner and outer surfaces. Therefore, it is feasible to reconstruct surfaces without using samples that are beneath the object surface. For this, we define a point set U with $f_o(U) = u$, $u \in [0, 1]$, where u indicates the possibility of a region being occupied in unexplored spatial regions. Then, we specify U as a 3D point grid within an interval distance $d_U \in \mathbb{R}^3$ (in our implementation, each entry in d_U is automatically calculated as $\frac{1}{10}$ of the corresponding dimension of the object's Axis-Aligned Bounding Box).

Next, we transform the tactile observations into training data. We denote the points in occupied regions as O , with $f_o(O) = 1$, and points in void regions as V , with $f_o(V) = 0$. To update the uncertainty set U according to the observation sets V and O , we remove the grid points in U that are close to points in V and O (i.e. for all p_u with $\|p_u - p_{vo}\|_2 < d$, $p_u \in U$ and $p_{vo} \in V \cup O$). The final training data is $\mathcal{D} = \{\chi, y\}$ with $\chi = O \cup V \cup U$ and $y = \{f_o(O) = 1\} \cup \{f_o(V) = 0\} \cup \{f_o(U) = u\}$.

We use Gaussian Process Regression [33] to learn the implicit function from the training data $\mathcal{D}$. We denote the training points in χ as $\mathbf{x} \in \mathbb{R}^3$, and test points as $\mathbf{x}_* \in \mathbb{R}^3$. The $f^*(\mathbf{x}_*) \in \mathbb{R}^1$, a posterior distribution of the occupancy function at $\mathbf{x}_*$, is subject to the a Gaussian function, which is described in (2)–(4),

$$p(f^*(\mathbf{x}_*) | \mathcal{D}) \sim \mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*)) \quad (2)$$

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \quad (3)$$

$$\sigma^2(\mathbf{x}_*) = k_{**} - \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (4)$$

Where k is the kernel function, k_{**} is a scalar calculated by $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$, $\mathbf{k}_*$ is the vector of covariances between the test point and the n training points, and $\mathbf{K}$ is a $n \times n$ matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i, \mathbf{x}_j \in \chi$. Using the regressed function f^* , the surface can be extracted by calculating the c -isosurface where $c \in [\min_{\mathbf{x}_i \in \chi} f^*(\mathbf{x}_i), \max_{\mathbf{x}_i \in \chi} f^*(\mathbf{x}_i)]$. In practice, we calculate c by the (5) with $\alpha = 0.05$.

$$c = \alpha \min_{\mathbf{x}_i \in \chi} f^*(\mathbf{x}_i) + (1 - \alpha) \max_{\mathbf{x}_i \in \chi} f^*(\mathbf{x}_i) \quad (5)$$

In addition, the surface normal at $\mathbf{x}_*$ can be estimated as $-\frac{\partial f^*(\mathbf{x}_*)}{\partial \mathbf{x}_*}$. We implemented the differential calculation by using the auto-differential function in GPyTorch [9].

C. Grasp Detection

The proposed grasp detection can be performed on two kinds of augmented tactile data: CAD model to align with tactile samples, or the implicit surface obtained from uGPIS. The latter case is only used when the minimum alignment error w.r.t all CAD models in the support set has exceeded the threshold. While the proposed framework is compatible with other types of grippers, our approach is tested on the gripper of the Taurus Dexterous Robot (SRI International). Based on the axis definition of the

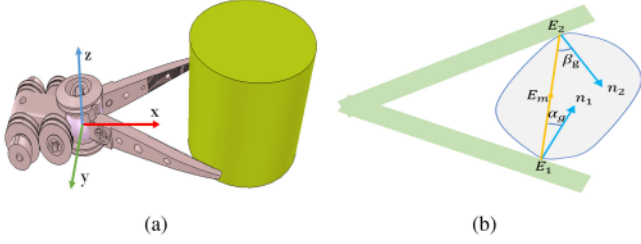


Fig. 4. (a) Axis definition of the gripper (b) Annotations of contact normals ($\mathbf{n}_1, \mathbf{n}_2$) and angles (α_g, β_g) in a planar case.

Algorithm 2: Grasp Detection For Taurus Gripper.

Input: P_{obj} : Object point cloud
 $P_{interest}$: Interest points on the object surface
 $N_{interest}$: Surface normals of $P_{interest}$
Result: B_{grasp} : A set of feasible grasps

```

1 for each  $\mathbf{p}_i, \mathbf{n}_i$  in  $(P_{interest}, N_{interest})$  do
2   Define Darboux Frame  $O_p$  at  $\mathbf{p}_i$ 
3   for  $\psi \in \Psi$  do
4     for  $\lambda \in \Lambda$  do
5       for  $d \in D$  do
6          $g = \text{Gripper}(O_p, \psi, \lambda, -d\mathbf{n}_i)$ 
7         if  $\text{Contact\_Filter}(g, P_{obj})$  then
8            $g' = \text{GraspRefine}(g, P_{obj})$ 
9            $B_{grasp} \leftarrow B_{grasp} \cup g'$ 
10        end
11      end
12    end
13  end
14 end
```

Taurus gripper (refer to Fig. 4), the grasp detection algorithm is given in Alg. 2.

1) *Generate Grasps*: The grasp is generated by performing exhaustive search on all the interest points $P_{interest}$ (e.g. points from Farthest Point Sampling [22]). The coordinate definition of the gripper is shown in the Fig. 4 (a). For point $\mathbf{p}_i \in P_{interest}$, we first align the gripper's x axis with the corresponding surface normal $\mathbf{n}_i \in N_{interest}$, and construct the Darboux Frame O_p to facilitate searching on the following parameters: the gripper's roll $\psi \in \Psi$, the gripper's open degree $\lambda \in \Lambda$, and the distance from the gripper's center to O_p : $d \in D$. For the gripper of the Taurus robot, we choose the parameter as: $\Psi = \{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}\}$, $\Lambda = \{\frac{\pi}{6}, \frac{\pi}{3}\}$ in rad, and $D = \{0.02, 0.04\}$ in meters.

2) *Grasp Refinement*: A grasp can be fully determined by a tuple of 4 elements: $(O_p, \psi, \lambda, -d\mathbf{n}_i)$. After generating a grasp g by the previous step, we filter out grasps that collide with the object or have low grasping quality (as explained below).

Our grasping quality metric is an expansion of the force closure condition [18], [25], [27] i.e. whether a grasp can resist external wrenches $\mathcal{F}_{ext}$ given the wrench basis $\mathcal{F}$, as shown in (6).

$$\mathcal{F} \cdot k + \mathcal{F}_{ext} = 0 \quad s.t. \quad k \geq 0 \quad (6)$$

Evaluating force closure condition in a spatial case requires three contact points. However, for our two-finger gripper, we can only

guarantee the existence of two contact points and therefore, we evaluate the grasps using *antipodal grasping theorem* [25] which defines a sufficient condition for the force closure condition. In this theorem, a spatial grasp with two soft-finger contacts is *force-closure* if and only if the line connecting the contact point lies inside both the friction cones [25]. Note that the *antipodal grasping theorem* gives a binary criterion. However, we want to quantify the grasping quality as a continuous numerical value to facilitate comparisons between different grasps. For this, we transform the antipodal grasping theorem into a likelihood scoring function as given in (7) where α_g and β_g are the angles between contact normals $\mathbf{n}_1, \mathbf{n}_2$ and the line E_1E_2 (Fig. 4(b)), and, $\cos(\alpha_g)$ and $\cos(\beta_g)$ are empirical likelihoods that line E_1E_2 lies within the two friction cones, respectively.

$$s = \cos(\alpha_g) \cdot \cos(\beta_g) \quad (7)$$

Nevertheless, recent studies suggest that the force closure condition is not a perfect grasp quality metric [40]. For this, we filter out grasps that are empirically not reliable using the following criteria. Note that the soft contact criterion and the mass distribution criterion are only applicable when the position of the center of mass is available.

Grasp Quality Criterion: Grasps with $s < 0.5$ are discarded. Further, both of the two angles α_g, β_g are required to be smaller than $\frac{\pi}{3}$ considering the limited friction coefficient.

Soft Contact Criterion: The soft finger contact may not generate sufficient torque to counteract $\mathcal{F}_{ext}$. We discard a grasp if the gravity center is more than 5 cm away from E_m (the midpoint of E_1E_2 , as depicted in Fig. 4 (b)), and if the spatial angle between $\mathbf{g} \times (\mathbf{E}_m - \mathbf{C}_m)$ and the y axis is smaller than $\frac{\pi}{3}$, where $\mathbf{C}_m$ is the position of object's center of mass.

Mass Distribution Criterion: The antipodal grasp theorem does not consider the object's mass distribution. In practice, balancing the mass distribution contributes to the grasp quality by avoiding the in-gripper rotation [5]. The gripper's xy plane splits the object's Oriented Bounding Box (OBB) into two regions. The grasp is discarded if the maximum volume ratio of two regions is larger than 3.

IV. EXPERIMENTS AND RESULTS

A. Tactile Sampling

A description of the meta-configuration of tactile sampling is given as follows. In the workspace W , there exists a tactile sensor probe and an unknown object J . The probe's position $\mathbf{p}_p$ can always be observed. When the probe has contact with J , the coordinates of the contact point set P_c and its normal vectors N_c can be detected.

Inspired by Ting *et al.* [51], we created an interactive simulation environment corresponding to this tactile sampling configuration, as shown in the Fig. 5 (a). The purple sphere attached to the gripper is the sensor probe. The object models used in this work for tactile sampling are a subset of objects presented in YCB dataset [2]. Fig. 5 (b) corresponds to our real experiment scenario, in which a Taurus robot is collecting tactile signatures for an object being concealed inside the soft plastic cover by pressing the regions of interest.

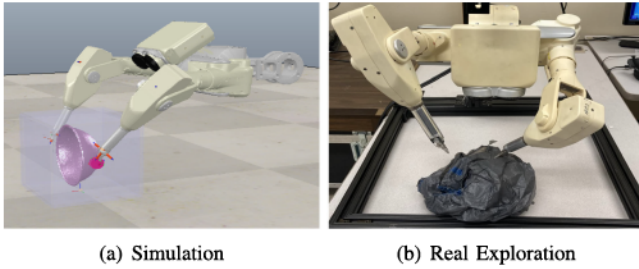


Fig. 5. (a) Simulation environment for interactively obtaining tactile samples. (b) Tactile exploration using Taurus dexterous telepresence manipulation system.

B. Object Recognition

This section addresses the problem of *inductive bias* in the classifier network which is used to propose candidate objects from a support set. To indicate the negative effects of *inductive bias*, we performed the classification task on the point cloud data obtained from tactile sampling. We compared the Triangle-Net approach with two other baselines that do not have robustness towards multi-factorial variances: PointNet [32] and DGCNN [43]. All approaches are trained with point cloud patches obtained from 24 objects that belong to the YCB dataset (512 points per object). The data variation is augmented by: 1) applying $SO(3)$ rotation to the objects, which simulates real scenarios where object pose is not known in advance, 2) using the proposed data augmentation (Alg. 1) to generate partial observations as it is expensive to obtain large amounts of human-guided tactile samples. The network is tested on YCB objects observed by interactive tactile exploration (Section IV-A) guided by a human (74 observations in total). To ensure sample diversity, each object is sampled with at least three observations from different sampling patterns. Some examples can be found in the Fig. 3. We down-sampled the test observations to 512 points per object since the accuracy would otherwise be affected by the number of points [45]. The mean test accuracies are reported in Table I.

We observe significant differences in performance between Triangle-Net and other methods. For top-1 accuracy, Triangle-Net achieves 47.9% while PointNet and DGCNN achieve only 19.2% and 15.1%, respectively. Similarly, this performance gap also exists in top-5 accuracy. This indicates that the robustness towards multi-factorial disturbances is crucial to reduce the inductive bias. To quantify the effectiveness of data augmentation, we show the degraded classification accuracy when the data augmentation method is removed (i.e. “w/o aug” in Table I, which trains network on YCB objects with $SO(3)$ rotation but without missing regions).

C. Surface Reconstruction

First, we show the necessity of uGPIS algorithm by visually comparing the reconstruction results with GPIS (Fig. 6). In this experiment, we analyze the tactile sampling scenario where the SDF value cannot be directly observed. For GPIS, the SDF value is estimated from the surface normal (refer to [19], [29]).

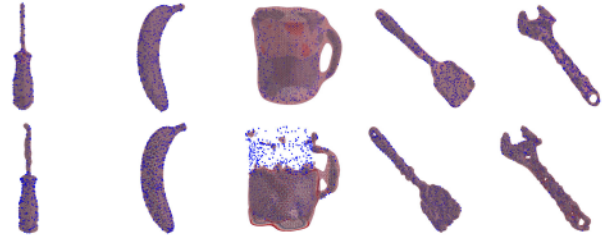


Fig. 6. Comparison of surface reconstruction results by uGPIS (row 1) and GPIS (row 2).

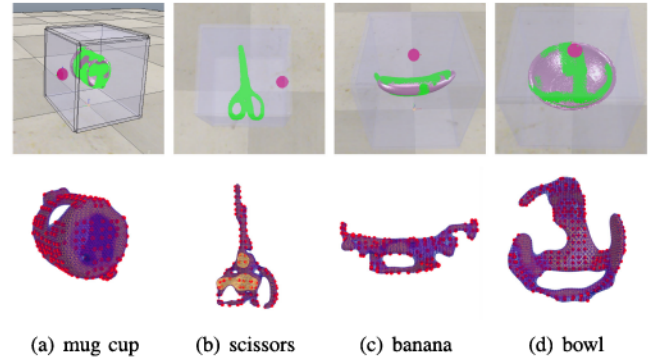


Fig. 7. High resolution implicit surface from uGPIS on non-watertight samples.

Since d_n in GPIS is a hyper-parameter that is related to the individual object scale, the parameter sets for GPIS have to be carefully tuned for each object, while uGPIS can robustly work on all objects using a unified set of parameters. Overall, the GPIS works well for watertight objects with simple geometries (e.g. a banana). However, the reconstructed surface is distorted as the estimated SDF is inaccurate when thin structures exist (e.g. a screwdriver, spatula, wrench), or completely fail when the watertight assumption does not hold (e.g. mug cup, as it can be touched from both inside and outside).

Next, we conduct experiments to show that uGPIS algorithm is capable of upsampling sparse and discrete tactile point clouds that are not considered to be watertight. The results obtained are shown in the Fig. 7. The first row depicts our tactile sampling environment, where the green regions are the contact regions made by the tactile probe. The second row shows the high-resolution implicit surface reconstructed by the uGPIS algorithm. The point cloud is shown in red color and the reconstruction surface is visualized in blue color. Results indicate that uGPIS can successfully represent the non-watertight and discrete contacts as continuous surfaces with infinite resolution.

D. Grasp Detection

Next, we compare the quality of the grasp detection under three scenarios: 1) Directly using the low-resolution and partially observed point clouds, 2) Using upsampled point clouds generated by uGPIS (16 384 points per object), and 3) Using the CAD models aligned with the partially observed objects. We evaluated the grasp detection algorithm on three objects from

TABLE II

GRASP QUALITY EVALUATION ON 3 SCENARIOS: (1) RAW TACTILE POINT CLOUDS, (2) UPSAMPLED POINT CLOUDS BY uGPIS, (3) FULL OBJECT MODELS

Data type	Object	① Total grasps	② Collision free grasps	③ Success rate	④ $③ \times ② / ①$
(1) Raw points	A	101	29	18/29	17.8%
	B	64	21	13/21	20.3%
	C	269	44	25/30	13.6%
(2) uGPIS	A	204	80	17/30	22.2%
	B	993	668	22/30	49.3%
	C	844	276	23/30	25.1%
(3) Full model	A	74	74	26/30	86.7%
	B	1816	1816	26/30	86.7%
	C	1085	1085	25/30	83.3%

the YCB dataset: (A) mug cup, (B) scissors, and (C) Phillips screwdriver. We built a simulation based on V-Rep [34] for evaluating the stability of the grasps. A grasp is successful if the object does not fall under gravity.

Table II shows the percentage of successful grasps in three aforementioned scenarios. We can draw three conclusions from this table. First, as the column ① and ② indicate, for all 3 objects, using raw tactile point cloud data for grasp detection yields an insufficient number of grasps due to the sparsity. Using upsampled point cloud obtained from uGPIS, or using fully aligned CAD models yields a significantly larger number of grasps. Second, column ① and ② also indicate that, grasp detection on partially observed point clouds yields a large number of grasps that may collide with the unobserved parts of the objects. This problem can be solved by using the completed models that are aligned with the tactile observations. In addition, using a full CAD model also facilitates grasp detection on unobserved parts. Third, our overall framework achieves superior performance in terms of success rate of collision-free grasps (column ③) and the overall success rate (column ④). Hence, we conclude that utilizing both the surface reconstruction from uGPIS and the fully observed models from the support set are beneficial to the quality of planned grasps, while the latter approach has more significant advantages.

E. Framework Evaluation

Next, we evaluated the performance of the whole framework. The support set has 14 CAD models chosen from the YCB dataset. The evaluation is conducted under two scenarios: 1) Grasp 14 objects from the support set, and 2) Grasp 10 objects from YCB dataset that do not belong to the support set. For each object, we collected three observations by tactile exploration (refer to Section IV-A). Since each observation is evaluated for three grasps, it results in 9 grasps for each object, 126 grasps for objects in the support set, and 90 grasps for novel objects. In these experiments, uGPIS is considered as a regression model and is trained with RBF kernel for 200 epochs at a learning rate of 0.1. We trained the Triangle-Net with the artificial data obtained from data augmentation (Alg. 1). we set $M \in [3, 4, 5]$, $r_t = 0.6$, $r_s = 0.4$, and for ICP, we registered the Top-5 objects proposed by the Triangle-Net.

TABLE III

EVALUATION RESULTS OF THE WHOLE FRAMEWORK

Object Type	Total grasps	Model-based grasps	Model-free grasps	Success rate
Support set objects	126	59/78	21/48	63.5%
Novel Objects	90	1/3	31/87	35.6%

The obtained results are shown in the Table III. In scenario 1 (support set objects), 78 observations are successfully registered to CAD models, resulting in 59 successful grasps. For the remaining 48 observations, the classification/registration algorithm cannot attribute it to the correct CAD model and thus, the grasp detection is performed based on the implicit surface reconstructed by uGPIS. For this, 21 of 48 grasps are successful. The overall success rate for support set objects is $\frac{80}{126} = 63.5\%$. In the second scenario (novel objects), we achieve 32 successful grasps out of 90 trials, showing the framework's generalization abilities to new or unseen objects. We notice a case where one observation of "small clamp" model is incorrectly registered to "large clamp" model due to partial similarity and triggered the model-based grasping. This issue can be avoided by adding more tactile points to distinguish an object from other candidates.

V. CONCLUSION

To tackle the intractability of object inference and planning manipulation policies based purely on the tactile signatures, we propose FIST-D, a tactile-based decision-making framework that augments tactile observations by integrating knowledge from a support set of CAD models. The main challenge of utilizing tactile point clouds is the inductive bias induced by the multifactorial variances in the tactile measurements. To address this issue, our proposed framework adaptively integrates two solutions: 1) An object-aware approach that proposes high-quality CAD objects from the support set, and 2) uGPIS, an occupancy possibility-based object reconstruction method that transforms discrete contact points into the implicit surface representation, enabling generalization to unseen objects. Further, we propose a simulation paradigm that allows tactile sampling to be conducted interactively. By utilizing the above framework, we managed to plan grasps directly based on tactile points. In the future work, we will relax the assumption that the configuration of the object needs to remain unchanged during tactile exploration, and develop techniques to tackle this challenge. This issue can be potentially addressed by using sensors that require low contact force such as a triboelectric sensor [47]. Lastly, there is a risk that the success rate may be hampered when the tactile measurements are very noisy. This can be mitigated by utilizing the recent advancements in high resolution tactile sensors such as GelSight [48].

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

REFERENCES

- [1] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
- [2] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," 2015, *arXiv:1502.03143*.
- [3] D. Campbell and L. Petersson, "An adaptive data representation for robust point-set registration and merging," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4292–4300.
- [4] M. Charlebois, K. Gupta and S. Payandeh, "Curvature based shape estimation using tactile sensing," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1996, pp. 3502–3507.
- [5] D. Chen, V. Dietrich, Z. Liu, and G. von Wichert, "A probabilistic framework for uncertainty-aware high-accuracy precision grasping of unknown objects," *J. Intell. Robot. Syst.*, vol. 90, no. 1/2, pp. 19–43, 2018.
- [6] D. Driess, D. Hennes, and M. Toussaint, "Active multi-contact continuous tactile exploration with gaussian process differential entropy," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 7844–7850.
- [7] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," 2020, *arXiv:2011.09584*.
- [8] W. Gao and R. Tedrake, "FilterReg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11095–11104.
- [9] J. Gardner, G. Pleiss, Kilian Q. D. Weinberger Bindel, and A. G. Wilson, "GPYtorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 7576–7586.
- [10] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11366–11374.
- [11] M. Kaboli, K. Yao, D. Feng, and G. Cheng, "Tactile-based active object discrimination and target object search in an unknown workspace," *Auton. Robots*, vol. 43, no. 1, pp. 123–152, 2019.
- [12] H. M. Kasaee, J. Melsen, F. van Beers, C. Steenkis, and K. Voncina, "The state of service robots: Current bottlenecks in object perception and manipulation," 2020, *arXiv:2003.08151*.
- [13] D. Kent, M. Behrooz, and S. Chernova, "Crowdsourcing the construction of a 3 d object recognition database for robotic grasping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 4526–4531.
- [14] D. Kent and S. Chernova, "Construction of an object manipulation database from grasp demonstrations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3347–3352.
- [15] D. Kragic and H. I. Christensen, "Model based techniques for robotic servoing and grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 299–304.
- [16] H. Liang, et al., "Pointnetgpd: Detecting grasp configurations from point sets," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 3629–3635.
- [17] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, pp. 54–67, 2017.
- [18] K. M. Lynch and F. C. Park, *Modern Robotics*, Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [19] J. Mahler et al., "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 4919–4926.
- [20] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 630–635, Jun. 2011.
- [21] T. Miyamoto, H. Sasaki, and T. Matsubara, "Exploiting visual-outer shape for tactile-inner shape estimation of objects covered with soft materials," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6278–6285, Oct. 2020.
- [22] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling. Technical report," Univ. Cambridge, Computer Laboratory, 2003.
- [23] A. Mousavian, C. Eppner, and D. Fox, "6 DOF GraspNet: Variational grasp generation for object manipulation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.
- [24] A. Murali, Y. Li, D. Gandhi, and A. Gupta, "Learning to grasp without seeing," in *Proc. Int. Symp. Exp. Robot.*, 2018, pp. 375–386.
- [25] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, Boca Raton, FL, USA: CRC press, 1994.
- [26] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.
- [27] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.
- [28] G. M. Nielson, "On marching cubes," *IEEE Trans. Visual. Comput. Graph.*, vol. 9, no. 3, pp. 283–297, Jul.–Sept. 2003.
- [29] S. Ottenhaus, M. Miller, D. Schiebener, N. Vahrenkamp, and T. Asfour, "Local implicit surface estimation for haptic exploration," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 850–856.
- [30] C. Pacchierotti, A. Tirmizi, and D. Prattichizzo, "Improving transparency in teleoperation by means of cutaneous tactile force feedback," *ACM Trans. Appl. Perception*, vol. 11, no. 1, pp. 1–16, 2014.
- [31] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50 k tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 3406–3413.
- [32] R. Qi, H. K. SuMo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3 d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [33] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer Sch. Mach. Learn.*, 2003, pp. 63–71.
- [34] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.
- [35] C. Rosales, F. Spinelli, M. Gabbicini, C. Zito, and J. L. Wyatt, "GPAtlas-RRT: A local tactile exploration planner for recovering the shape of novel objects," *Int. J. Humanoid Robot.*, vol. 15, no. 01, 2018, Art. no. 1850014.
- [36] J.-P. Saut, S. Ivaldi, A. Sahbani, and P. Bidaud, "Grasping objects localized from uncertain point cloud data," *Robot. Auton. Syst.*, vol. 62, no. 12, pp. 1742–1754, 2014.
- [37] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *Int. J. Robot. Res.*, vol. 15, no. 3, pp. 230–266, 1996.
- [38] E. J. Smith, et al., "Jitendra malik, and michal drozdal. 3 d shape reconstruction from vision and touch," 2020, *arXiv:2007.03778*.
- [39] R. F. Solodova et al., "Instrumental tactile diagnostics in robot-assisted surgery," *Med. Devices (Auckland, NZ)*, vol. 9, pp. 377, 2016.
- [40] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, no. 13/14, pp. 1455–1473, 2017.
- [41] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2442–2447.
- [42] S. Wang et al., "3 D shape perception from monocular vision, touch, and shape priors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1606–1613.
- [43] Y. Wang et al., "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
- [44] O. Williams and A. Fitzgibbon, *Gaussian Process Implicit Surfaces*. 2006.
- [45] C. Xiao and J. Wachs, "Triangle-Net: Towards robustness in point cloud learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 826–835.
- [46] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: Can we turn tactile sensing into vision?," *Adv. Robot.*, vol. 33, no. 14, pp. 661–673, 2019.
- [47] H. Yang, F. R. Fan, Y. Xi, and W. Wu, "Bio-derived natural materials based triboelectric devices for self-powered ubiquitous wearable and implantable intelligent devices," *Adv. Sustainable Syst.*, vol. 4, no. 9, 2020, Art. no. 2000108.
- [48] W. Yuan, S. Dong, and E. H. Adelson, "GelSight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017, Art. no. 2762.
- [49] M. Zhang, *Tactile Perception and Visuotactile Integration for Robotic Exploration*. 2019.
- [50] M. M. Zhang, N. Atanasov, and K. Daniilidis, "Active end-effector pose selection for tactile object recognition through monte carlo tree search," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 3258–3265.
- [51] T. Zhang, T. Zhou, Bradley S. Duerstock, and J. P. Wachs, "Image exploration procedure classification with spike-timing neural network for the blind," in *Proc. 24th Int. Conf. Pattern Recognit.*, 2018, pp. 3256–3261.