# Simple Kinesthetic Haptics for Object Recognition

Avishai Sintov and Inbar Meir*

## Abstract

Object recognition is an essential capability when performing various tasks. Humans naturally use either or both visual and tactile perception to extract object class and properties. Typical approaches for robots, however, require complex visual systems or multiple high-density tactile sensors which can be highly expensive. In addition, they usually require actual collection of a large dataset from real objects through direct interaction. In this paper, we propose a kinesthetic-based object recognition method that can be performed with any multi-fingered robotic hand in which the kinematics is known. The method does not require tactile sensors and is based on observing grasps of the objects. We utilize a unique and frame invariant parameterization of grasps to learn instances of object shapes. To train a classifier, training data is generated rapidly and solely in a computational process without interaction with real objects. We then propose and compare between two iterative algorithms that can integrate any trained classifier. The classifiers and algorithms are independent of any particular robot hand and, therefore, can be exerted on various ones. We show in experiments, that with few grasps, the algorithms acquire accurate classification. Furthermore, we show that the object recognition approach is scalable to objects of various sizes. Similarly, a global classifier is trained to identify general geometries (e.g., an ellipsoid or a box) rather than particular ones and demonstrated on a large set of objects. Full scale experiments and analysis are provided to show the performance of the method.

## 1 Introduction

When humans reach for an object, such as a mobile phone or the computer mouse, they can identify the object even while their gaze is directed elsewhere (Lederman and Klatzky, 1993). Some finger contact with the object is usually sufficient to acquire essential information to instantly identify the object (Klatzky et al., 1985; Klatzky and Lederman, 1995). The notion of *Exploratory procedures* (EP), defined by Lederman and Klatzky (1987), hypothesized that haptic exploration typically involves a stereotypical set of hand gestures, each aimed to identify a specific property of the object. These include lateral motion, pressure and some arbitrary contact for texture, stiffness and temperature sensing. More relevant topics are finger enclosure and contour following for shape or volume recognition (Driess et al., 2017).

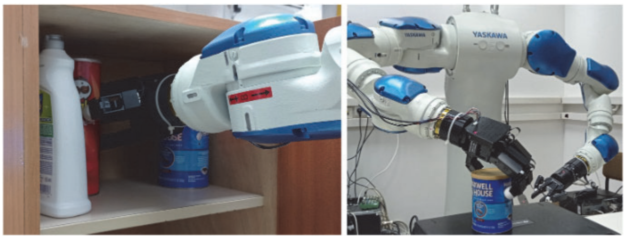


Figure 1: Robotic haptic glances to identify (left) an occluded object in a cabinet and (right) on a table.

Inspired by these studies, robotic hands have been developed to include haptic capabilities, enabling the identification of objects and their properties (Dario et al., 1992). Contrary to visual sensing, haptic sensors are capable of directly perceiving physical properties of an object such as shape, compliance, texture and temperature. With haptic sensing technology, robots can extract information of an object even when the object is fully or partially occluded to a visual system. This can happen when, for instance, searching for an object at the back of the closet or in a box full of items.

Information from haptic sensors is acquired through direct interaction with objects usually involving both tactile sensing and internal sensing of joint actuators known as Kinesthetic haptics (Carter and Fourney, 2005). Traditionally, tactile refers to information received from touch and contact sensing, while kinesthetic refers to information sensed through movement, force or position of joints and actuators. Tactile sensing with or without visual perception is the leading method for haptic-based object recognition (Rouhafzay and Cretu, 2020). The current state of the art in tactile sensing for identifying and classifying objects typically uses multiple high-density tactile sensors on a robotic hand. However, this can be expensive and may complicate

[*] Sintov and Meir are with the School of Mechanical Engineering, Tel-Aviv University, Israel.

the hardware, especially in already complex multi-finger hands (Drimus et al., 2011). Furthermore, training classifiers to process tactile information requires actual data collection from a set of objects (Spiers et al., 2016; Lin et al., 2019; James et al., 2020). Hence, even if tactile sensors exist, actual collection is required and may not be transferable to another hand. Simulating viable tactile signals, on the other hand, is a challenging task and cannot be used straight-forward to train a classifier (Narang et al., 2020). In addition, most of the existing work can only deal with simple shaped objects (Liu et al., 2017). Previous work on object recognition without tactile sensors suggested to map the surface of the object by collecting a point cloud of contact points using a robotic hand (Allen and Roberts, 1989; Jin et al., 2013). The data is then used to fit a parametric surface model. The method, however, requires a large dataset of actual measurements and simplifies the model to some quadratic form.

In our work, we propose a kinesthetic method to classify objects with neither tactile sensors nor visual perception, and without data collection on actual objects. A grasp embeds an instance of the surface of held objects. Different objects have different patterns of grasps based on their shapes. Such grasp, however, is usually described by the location of the contact points and depends on the used reference frame. Hence, a grasp described by the contact points is not a unique representation and may be challenging when used to train a classifier. On the other hand, a grasp parameterization technique proposed by Sintov et al. (2014, 2016) is unique and independent of any coordinate frame. The proposed representation treats a spatial grasp as a polyhedron and provides an injective parameterization. One may add more descriptive information of the object by also including the normals at the contact points relative to the polyhedron.

In this paper, we show that with the proposed grasp representation, a classifier can be trained to identify complex objects within several grasps. The classifier is trained solely on object information based on its geometry without the need of a physics-engine simulator. The classifier is independent of any particular robot hand and, therefore, can be applied to various ones. A grasp, in this context, is an *haptic glance* (Dragiev et al., 2013) aimed to gain more information and improve the classes probability distribution. Such grasp is demonstrated by the Allegro hand in Figure 1. Generating training data is solely done computationally by sampling contact points from the surface of Computer-Aided Design (CAD) models of the objects independent of the type of the used hand. Actual collection of data with a real robotic hand and real objects is not required. Hence, adding an object is fast (up to several minutes). If a CAD model is not available, the object can be scanned or actual data collection with a robotic hand is possible. Actual grasp information for object recognition queries is acquired through kinesthetic sensing, i.e., measuring joint angles and torques of the fingers while grasping the query object.

We assume that the general location of the object is a priori known. The location of the object can be acquired through low-level object localization without complex Convolutional Neural Network classification (Du et al., 2020). For instance, 2D salient object detection can be applied through visual perception with a simple camera either on the hand or around the robot. We leave the object localization problem out of the scope of this paper. In this work, we consider robotic hands of two or more fingers. While simple grippers like suction cups provide good picking-up performance, they cannot use a picked-up object to perform a task (e.g., use a drill or pour from a bottle). Hence, the use of a multi-finger hand is motivated by the necessity for dexterity in order to perform real-world tasks (OpenAI et al., 2019; Morgan et al., 2022; Azulay et al., 2023). With the proposed approach, the robot can identify an object upon pick-up with an intent to utilize it to perform a task.

When classifying an object, one grasp generally does not include enough surface information. Therefore, we propose and investigate the use of iterative methods to improve predictions, and provide a comparative analysis between methods. An illustration of the process is shown in Figure 2. We propose to use either a scoring-based iterative method or Bayesian update to approximate the certainty about an object based on a few grasp samples. We also investigate the ability to classify an object with partial grasp information solely with finger contact location while not including the normals at the contact points. Furthermore, we explore the ability of the technique to generalize to families of objects (e.g., ellipsoids, cylinders) and not only specific objects with available CAD models. Hence, we provide a global classifier for any multi-finger robotic hand to recognize general types of objects. We also exhibit a scale invariant property of the method when simply normalizing the grasp polyhedron.

To summarize, the contributions of this paper are as follows. An iterative classification approach is proposed for recognizing an object grasped by a multi-fingered hand. The classification is based on kinesthetic-based haptic glances where only the kinematics of the hand are required. The method utilizes a simple parameterization which provides a unique grasp representation independent of any coordinate frame. Hence, the approach does not depend on a specific object placement during sampling. As tactile sensors are hard to simulate, training an object recognition model would require

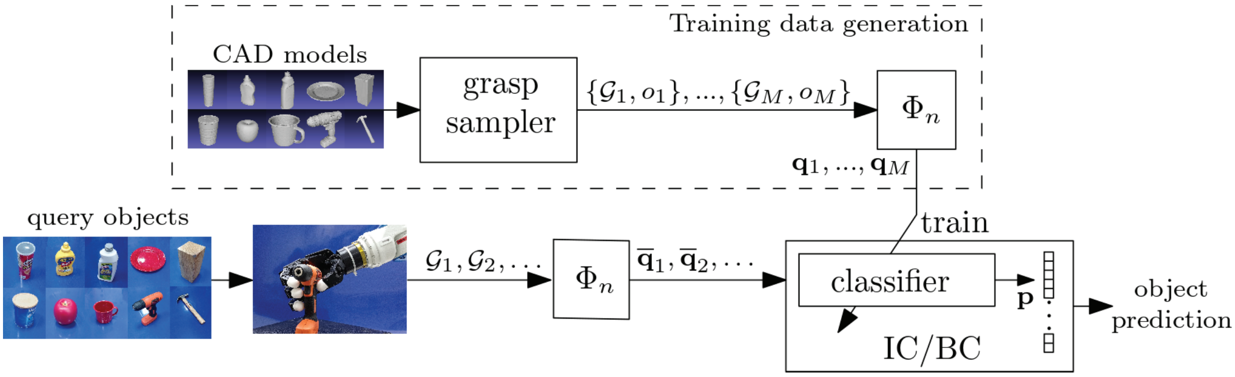

Training data generation
CAD models
grasp sampler
$\{\mathcal{G}_1, o_1\}, ..., \{\mathcal{G}_M, o_M\}$
$\Phi_n$
$\mathbf{q}_1, ..., \mathbf{q}_M$
train
query objects
$\mathcal{G}_1, \mathcal{G}_2, ...$
$\Phi_n$
$\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2, ...$
classifier
$\mathbf{p}$
IC/BC
object prediction

Figure 2: Illustration of the (top) training and (bottom) evaluation processes. CAD models of the objects are used to generate labeled data for training a classifier. The classifier, trained solely over CAD models and independent of any particular robot hand, can now be used to classify real object. Recognition of real objects is done by random sampling grasps using a robotic hand in an iterative classification process - Iterative Classification (IC) or Bayesian Classification (BC).

an extensive amount of real-world samples. In contrary, in the proposed approach data is generated in a computational process without the need for real-world grasp data. While the method is tested in this work on two robotic hands, the approach is expected to work on any multi-finger hand where the kinematics are known. The method is shown to generalize to families of objects and for objects of different scale.

## 2 Related work

Humans and animals use haptic sensory receptors and EP to perceive and interact with the environment (Nelinger et al., 2016). A study has identified neurons in the post-central gyrus of monkeys that are activated selectively to grasped objects while not activated when touching their own body (Iwamura et al., 1995). These neurons are tuned to specific features that discriminate common and familiar objects from the animal's own body. Inspired by these studies, robotic systems have been developed to grant a robotic hand with such haptic capabilities and to identify objects. These efforts will now be discussed.

### 2.1 Object Learning through tactile perception

Tactile perception in robotics is commonly used to learn a priori unknown object in uncertain environments to grasp and manipulate it. The learning is generally used to either learn features of an object or for object recognition. The latter is discussed in the next section. Feature learning may include compliance (Su et al., 2012), texture (Fishel and Loeb, 2012; Yuan et al., 2017), temperature variations (Dario et al., 1992) and surface modeling. A common practice to model the surface of an unknown object is by iteratively selecting discrete locations, known as *haptic glances*, where a robot touches (Dragiev et al., 2013; Yi et al., 2016). Tactile glances are used to improve an underlying surface model of the object using Gaussian process regression while incorporating model uncertainty. Similarly, tactile glances were used to augment noisy visual perception of an object (Björkman et al., 2013). In the tactile glances process, Jamali et al. (2016) proposed to incorporate a Gaussian process classifier to identify the boundary of the object and to filter-out glances that do not belong to it. More recently active exploration was proposed by Driess et al. (2017) where a robotic arm slides a probe over the surface of the object to acquire surface information. By exploring paths over the object, a continuous approximation of its surface is acquired.

### 2.2 Object Recognition

Object recognition has been widely researched in the past few decades and can be divided into methods of visual (Lowe, 1999) and tactile (Okamura et al., 1997; Bhattacharjee et al., 2012) perceptions. Both aim to identify an object in the vicinity of a robot. For visual perception, deep neural networks, or convolutional neural networks (CNN) in particular, have been making significant achievements in detection and classification of objects seen in 2D images (Deng et al., 2009; Lin et al., 2014). In contrary, classification based on 3D (depth) images is usually much more complicated due to higher dimensional representation of the data and the required computational effort (Griffiths and Boehm, 2019). PointNet is a recent advancement where a unified neural network directly takes raw 3D point clouds for object classification and segmentation (Charles et al., 2017). In another work, a set of 2D images taken from a 3D CAD model was rendered to generate a multi-view CNN that classifies objects from different viewing angles (Su et al., 2015). While these methods achieve high success rates, they require a substantial computational power and a direct line of sight with an object limited to a viewing area that obscures its back-facing region. The object can also be fully or partly occluded. In another approach, noise was used to recognize objects by tapping on them (Torres-Jara et al., 2005) or making noisy manipulation motions with the objects like shaking, dropping or crushing (Sinapov et al., 2011).

Several approaches learn to complete noisy or partial 3D point clouds of object shapes (Han et al., 2017; Zhou et al., 2021). Recently introduced, ShapeFormer is a transformer-based model that can output a 3D shape of an object based on partial depth information (Yan et al., 2022). The output is a distribution of possible outcomes that do not necessarily match the real object. In addition and despite high performance, shape completion methods require that a large portion of the object be visible. Such information is not always accessible in real-world tasks. For instance, grasping objects at the back of a cabinet or in confined spaces may not allow a sufficient line-of-sight. Shape completion approaches, however, can be used to refine 3D scans of objects in order to obtain a CAD model prior to training a classifier proposed in this paper. While a visual approach may exhibit good results in object recognition, relying on continuous visual feedback can limit the performance of various tasks in which visual uncertainty (e.g., poor lighting or shadows) or occlusion may occur. Achieving haptic-based object recognition would enable further fusion with vision for a complete recognition solution in unstructured or partly-occluded environments.

Contrary to vision, tactile-based recognition requires direct contact with an object. Supervised learning techniques are the preferred approach to map tactile feature data from an object to its class. Hence, the common research question lies in the type of data to measure. Drimus et al. (2014) developed flexible tactile sensors that provide pressure images during object squeezing. Tactile information corresponding to the pressure distribution during the squeezing was used with a nearest neighbor classifier to classify the object in hand. Luo et al. (2018) used patterns of tactile images along with kinesthetic information while utilizing $k$-means clustering to increase classification success rate. Similar tactile technology described the local shape of objects by the covariance matrix of the pressure image and used in a Naive Bayes classifier (Liu et al., 2012). Another work by Rouhafzay and Cretu (2020) combined visual perception to identify salient points over the surface of the objects in order to improve object recognition from a haptic image glance.

Spiers et al. (2016) proposed a method to perform a single grasp of an object using an underactuated hand with embedded pressure sensors on each finger. A hybrid approach uses measured pressure data and a hand kinematic estimator to train a random forest classifier and a parametric method. The approach enables to determine the class of an object and its physical properties including stiffness and size. All of the above tactile methods require somewhat tedious actual collection of data for all objects prior to training a classifier. Furthermore, these methods make use of complex, high-density sensor systems on the robotic hand which can be highly expensive and complicate the hardware. Our proposed approach, however, is distinguished from previous work by not requiring any tactile sensors and by the ability to train a classifier solely with data generated computationally on CAD models of the objects.

The work by Allen and Roberts (1989) has some similarities to ours in which a large dataset of contact points is used to fit a quadratic surface model and further classify the shape. Further work included also normal information in the point cloud of contacts (Jin et al., 2013). Both methods simplify the query objects to a set of shape primitives such as spheres and cylinders. We, however, use the full representation of an object, do not reconstruct its surface and use a significantly lower amount of data to identify it.

## 2.3 Grasp & Object Parameterization

Object shape parameterization has been widely researched for various applications (Sarkar et al., 2017). Much work has been done in the area of 3D shape similarity comparison. The work described by Museros et al. (2012), Novotni and Klein (2001) and Osada et al. (2002) applied algorithms that were, to that point, used for Internet and local database search, face recognition, image processing, or parts identification. The work of Ohbuchi et al. (2002) on shape similarity search uses a generalized feature vector of a 3D polygonal mesh constructed from the moment of inertia, average distance of the surface from the model's axis, and its variance. However, such methods deal with mean parameterization of the geometry (such as volume, shape distribution, moment of inertia) of the objects and cannot be applied for grasping.

Inspired by the above, Li and Pollard (2005) combined the notion of object parameterization with grasping. The work is based on shape matching for finding the best grasp for a set of objects. The best grasp is found by matching hand poses from a database of objects and human grasp postures. This is done by using a predefined parameterization of the object surface and the hand poses. An entire representation of any $n$-finger grasps was later presented that uniquely parameterized polygons formed by planar grasps (Sintov et al., 2012, 2014) and polyhedrons by spatial ones (Sintov et al., 2016). Such parameterization was used to compare grasps of different objects to find a common grasp and the matching design of a simple non-dexterous gripper to grasp them all. The method was also applied for grasping sheet-metal parts (Sintov and Shapiro, 2015). This parameterization is used in this work and presented next.

# 3 Grasp representation

In this section, we discuss a grasp representation based on previous work proposed by Sintov et al. (2014, 2016). An $n$-finger grasp of object $\mathcal{O}$ can be defined by a set of $n$ contact points,

$$\mathcal{P} = \left\{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^D \text{ for } i = 1, ..., n\right\} \tag{1}$$

on the object's surface, and the normal to the object's surface at each point

$$\mathcal{N} = \left\{\mathbf{n}_i : \mathbf{n}_i \in \mathbb{R}^D \text{ for } i = 1, ..., n\right\}, \tag{2}$$

where $\|\mathbf{n}_i\| = 1$ and, $D = 2$ or $D = 3$ for planar and spatial objects, respectively. We require a map $\Phi_n$ that takes any grasp $\mathcal{G} = \{\mathcal{P}, \mathcal{N}\}$ defined in some reference frame and maps it into a $w$-dimensional injective vector $\mathbf{q} \in \mathcal{Q}$ representing the grasp, where $\mathcal{Q} \subset \mathbb{R}^w$. That is, we require a map

$$\Phi_n : \mathcal{G} \to \mathcal{Q}, \tag{3}$$

such that $\Phi_n^{-1}(\mathbf{q}_j) \neq \Phi_n^{-1}(\mathbf{q}_k)$ for any $j \neq k$. In other words, we require a map that uniquely defines a grasp of an object and invariant to any coordinate frame. We note that the grasp representation to be described is valid for $n \geq 3$ while we will further exclude an $n = 2$ finger grasp case.

While the planar case is less likely to exist in the real world, for intuition and generality, we first describe a planar representation where $D = 2$. An $n$-finger planar grasp can be presented by an $n$-vertex polygon, where each vertex is positioned at a contact point as seen in Figure 3. Such polygon can be generated by constructing concave or convex polygons from a set of points in a plane according to polygonization algorithms (Deneen and Shute, 1988). As shown by Sintov et al. (2014), an $n$-vertex polygon requires $2n - 3$ constraints to define its shape and size. Explicitly, $n - 1$ internal angles of the polygon $\{\gamma_1, ..., \gamma_{n-1}\}$ along with $n - 2$ edge lengths $\{d_1, \ldots, d_{n-2}\}$ sufficiently define its shape and size summing up to $2n-3$ parameters. Further, to describe each contact, we define the normal at the contact relative to the grasp polygon, i.e., the normal direction can be defined by its angle relative to an adjacent edge of the polygon yielding $n$ additional parameters $\{\theta_1, \ldots, \theta_n\}$. The resulting $w$-dimensional grasp vector is

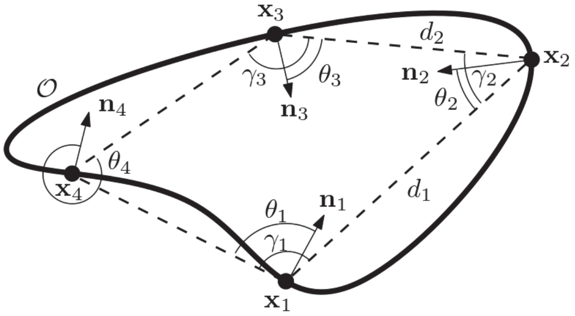


Figure 3: Example of a 4-vertex polygon describing a 4-finger ($n = 4$) grasp of object $\mathcal{O}$ with a vector of nine parameters.

$$\mathbf{q} = (\gamma_1, ..., \gamma_{n-1}, d_1, \ldots, d_{n-2}, \theta_1, \ldots, \theta_n) \tag{4}$$

where $w = 3n - 3$. For example, a 3-finger grasp will create a triangle parameterized with two angles and an edge length. Defining the normals will add three more angles yielding a 6-dimensional parameterization vector given by

$$\mathbf{q} = (\gamma_1, \gamma_2, d_1, \theta_1, \theta_2, \theta_3). \tag{5}$$

Similarly, the 4-finger grasp seen in Figure 3 forms a 9-dimensional vector. Such parameterization is not yet injective since several parameter choices can be made for the same polygon. Therefore and as a convention, parameter $d_1$ is always taken to be the longest edge while $d_2, ..., d_{n-2}$ are taken sequentially in the counter-clockwise direction.

In the spatial case where $D = 3$, the $n$ contact points $\mathcal{P}$ form a polyhedron $\Theta(\mathcal{P})$ with $n$ vertices as seen in Figure 4. A triangulation of a set $\mathcal{P}$ is the creation of simplices, where their vertices are points of $\mathcal{P}$ such that the union of them equals $\Theta(\mathcal{P})$ (Avis and El-Gindy, 1987). Triangulation of $\mathcal{P}$ can be done with algorithms such as the Quickhull (Bradford-Barber et al., 1996) or the space sweep technique (Preparata and Shamos, 1985). The algorithm will output a set of triangles forming $\Theta(\mathcal{P})$. Similar to the planar case, we divide the parameterization vector into constraints for the shape and size, and the ones that define the contact normals. An $n$-vertex polyhedron requires $3n - 6$ constraints to define its shape and size (Sintov et al., 2016). Here also, we may paramterize some of the triangles forming the polyhedron and their respected posture. However,

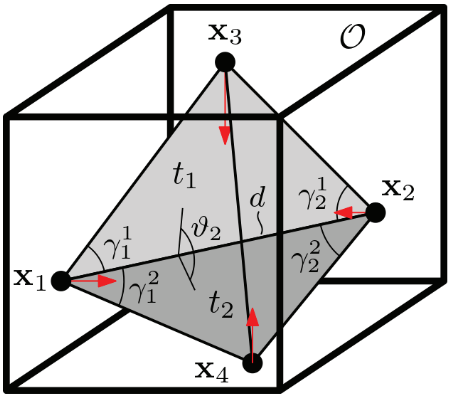


Figure 4: Example of a 4-finger grasp of a box. A 4-vertex polyhedron formed by four contact points is defined by six parameters. The contact normals are marked in red while their eight parameterization angles are omitted for simplicity.

there are many possible representations for the polyhedron depending on the order in which we parameterize its triangles. Hence, we parameterize the polyhedron in a specific order, ensuring an injective grasp representation. For brevity, we solely provide a brief overview of the algorithm while a full description is presented in previous work (Sintov et al., 2016). The algorithm first parameterizes the triangle (similar to the planar case - two angles $\gamma_1, \gamma_2$ and the longest edge length $d$) with the largest area, denoted as $t_1$, followed by the adjacent one $t_2$ through the longest edge. The parameterization moves on through a chain of $n-2$ adjacent triangles $t_j$, each with 3 parameters: two angles $\gamma_1^j, \gamma_2^j$ and the angle $\vartheta_j$ between triangle facets $t_j$ and $t_{j-1}$. Moreover, the algorithm describes the normals at the contact points relative to the polyhedron. Normal $\mathbf{n}_k$ needs two constraints in order to be defined and are given by two angles $\phi_1^k, \phi_2^k$ between the normal and triangles $t_1$ and $t_2$. For the 3-finger case, the two angles are defined relative to $t_1$ and the adjacent edge. Therefore, the resulting vector is of dimension $w = 5n - 6$ and is given by

$$\mathbf{q} = (\gamma_1^1, \gamma_2^1, d, \gamma_1^2, \gamma_2^2, \vartheta_2, \ldots \\ \gamma_1^{n-2}, \gamma_2^{n-2}, \vartheta_{n-2}, \phi_1^1, \phi_2^1, \ldots \phi_1^n, \phi_2^n). \quad (6)$$

For example, the 4-finger grasp in Figure 4 requires a 14-dimension vector: six for the geometry and eight for the normals:

$$\mathbf{q} = (\gamma_1^1, \gamma_2^1, d, \gamma_1^2, \gamma_2^2, \vartheta_2, \phi_1^1, \phi_2^1, \ldots \phi_1^4, \phi_2^4). \quad (7)$$

Map $\Phi_n$ presents a parameterization of a grasp described by contact points on the object and the normals at the contacts. In both planar and spatial cases, the map provides a unique vector form for a grasp. It should be noted that this grasp representation is unique only when normals are included. We now argue that such mapping is also invariant to any coordinate frame.

**Theorem 1.** *Let object $\mathcal{O}$ be described relative to coordinate frame $\mathcal{C}_i$, and let coordinate frame $\mathcal{C}_j$ be defined such that $\mathbf{c}_i = R\mathbf{c}_j + \mathbf{d}$ for any $\mathbf{c}_i \in \mathcal{C}_i$ and $\mathbf{c}_j \in \mathcal{C}_j$, and for some $R \in SO(3)$ and $\mathbf{d} \in \mathbb{R}^3$. Given a grasp $\mathcal{G}_k = \{\mathcal{P}_k, \mathcal{N}_k\}$ of the object described in $\mathcal{C}_i$, it must be that*

$$\Phi_n(\mathcal{G}_k) = \Phi_n(\mathcal{G}_k^{'}) \quad (8)$$

*with $\mathcal{G}_k^{'} = \{R\mathcal{P}_k + \mathbf{d}, R\mathcal{N}_k\}$.*

*Proof.* See Appendix B. □

Theorem 1 states that the above grasp representation is independent of any coordinate frame. Contact locations along with their normals are described relative to each other by a series of lengths and angles. The grasp representation is also not dependent on any robotic hand but only on contacts.

Measuring finger joint torques is not always possible or accurate enough and, therefore, the normals $\mathcal{N}$ at the contact points are not always available. In such case, only the contacts are available and the grasp is partially defined by $\mathcal{G} = \{\mathcal{P}\}$. The corresponding map $\Phi_n$ outputs a vector $\mathbf{q}$ of size $3n - 6$ only parameterizing the polyhedron formed by the contacts as previously described. In the above grasp parameterization, we have excluded 2-finger grasps ($n = 2$) where the normals at the contact points have no unique parameterization. Hence, 2-finger grasps can only be parameterized with no normals by only considering the distance between the contacts (one dimensional representation). We next use the above grasp parameterization for object recognition.

## 4 Object Recognition

In this section, we use the grasp representation described in Section 3 to identify a grasped object. Given a set of $m$ objects $\{\mathcal{O}_1, \ldots \mathcal{O}_m\}$, we require to identify an object from the set without any use of visual feedback or tactile sensors. The query object will be identified solely by placing finger tips of a robotic hand on the surface of the object or grasping it. For a recognition to be feasible, the objects should have some shape difference. For instance, two cups with the same general shape can be distinguished if only one has a handle. Although not addressed in this work, we note that including tactile sensors can be used to estimate normal forces instead of relying on hand kinematics.

It is assumed that all grasped objects are rigid. Hence, deformations between sequential grasp samples cannot occur making the samples mutually independent. Since the above grasp parameterization is independent of any coordinate frame, there is no assumption on the placement or orientation of the object during the sampling of a grasp. It is assumes that the general location of the object is a priori known. Furthermore, we assume that the robotic hand has access to joint angles, and its kinematics is known. Joint torques, however, are not always accessible in some robotic hands. Hence, we address both cases in which joint torques can and cannot be measured. It is also important to note that during a grasp sample, other parts of the fingers may touch the object. However, classification is based solely on contact locations of the fingertips.

Let $\theta_{f_i}$ and $\mathbf{x}_{f_i} \in \mathbb{R}^n$ be the vectors of finger joint angles and end-tip position of finger $i$. The forward-kinematics of the hand is given by

$$\mathbf{x} = \lambda(\theta) \tag{9}$$

where $\theta = (\theta_{f_1}^T, \ldots, \theta_{f_n}^T)^T$. Vector $\mathbf{x} = (\mathbf{x}_{f_1}^T, \ldots, \mathbf{x}_{f_n}^T)^T \in \mathbb{R}^{3n}$ is the concatenation of the finger tip coordinates relative to some global frame $\mathcal{H}$ on the hand. Furthermore, let $\tau_{f_i}$ be the vector of joint torques of finger $i$. Given joint torques vector $\tau = (\tau_{f_1}^T, \ldots, \tau_{f_n}^T)^T$ for the hand, the forces exerted by the fingers on the objects at the contacts are given by

$$\mathbf{f}_c = \left(J_h(\theta)^\dagger\right)^T \tau \tag{10}$$

where $J_h(\theta)$ is the hand Jacobian, $J_h(\theta)^\dagger$ is the pseudo-inverse of $J_h(\theta)$ and $\mathbf{f}_c = (\mathbf{f}_{f_1}^T, \ldots, \mathbf{f}_{f_n}^T)^T \in \mathbb{R}^{3n}$ is the concatenation vector of the forces (Murray et al., 1994). The force vectors are also expressed relative to $\mathcal{H}$ and can easily be transformed to the vector normals according to $\mathbf{n}_i = \mathbf{f}_{f_i}/\|\mathbf{f}_{f_i}\| \in \mathcal{N}$. The result is a representation of the current grasp $\mathcal{G} = \{\mathcal{P}, \mathcal{N}\}$ or $\mathcal{G} = \{\mathcal{P}\}$, when joint torques can and cannot be accessed, respectively. A grasp can now be mapped to the parameterization vector $\mathbf{q}$ and used as an input to the classifier to identify the object.

To generate a training set, we collect samples from CAD models of the objects. A model of object $\mathcal{O}_l$ can be defined by a set of tuples $\mathcal{F}_l = \{(\mathbf{x}_1, \mathbf{n}_1), (\mathbf{x}_2, \mathbf{n}_2), \ldots\}$ defining points and corresponding normals of the object surface. The number of contacts $n$ is defined by the number of fingers in the available robotic hand. We note, however, that grasp samples are collected by only picking contact locations on the objects independent of the kinematics of some hand. For each object, we collect $N$ samples of $n$-finger grasps of the form $\mathcal{G}_i$ by sampling uniformly at random of $n$ tuples from $\mathcal{F}_l$. Each sample is then mapped to a parameterization vector through $\mathbf{q}_i = \Phi_n(\mathcal{G}_i)$. The vector $\mathbf{q}_i$ is labeled $o_i \in \{1, \ldots, m\}$ corresponding to object $\mathcal{O}_{o_i}$. To enable tactile-like recognition, we do not require force-closure grasps but only that all fingertips are on the surface of the object. Thus, we may sample both force-closure and non-force-closure grasps. It is possible, as will be shown in experiments, to add noise to the CAD models in order to make the classifier robust to inaccurate models and finger-tip measurement errors.

The outcome of the above process is a set of observed data $Q_M = \{(\mathbf{q}_1, o_1), \ldots, (\mathbf{q}_M, o_M)\}$, where $M = mN$. Dataset $Q_M$ can now be used to train a classifier for a set of objects, i.e., train map $h : \mathcal{Q} \to [0,1]^m$ such that a class probability distribution $\mathbf{p} = \{p_1, \ldots p_m\}$, where $\sum_m p_i = 1$, is computed by $\mathbf{p} = h(\mathbf{q})$. Once a classifier is trained and validated, the identification of an object can be performed by sampling finger positions on the object. However, the parameterization described above only represents a grasp and does not include the entire information of an object. Furthermore, measuring finger joint torques is not always possible or accurate enough such that information of a grasp can be partial (two different grasps can have the same finger positions $\mathcal{P}$). Therefore, a classifier, that is trained with complete information of a grasp or solely on finger position representation, may yield object recognition with low certainty. Nevertheless, this could be improved by sequentially drawing more samples.

While common classification tasks rely only on one sample for prediction, in this case, we may acquire additional samples while being certain that they originate from the same class. Consider grasp vectors arriving sequentially $\{\mathbf{q}_1, \mathbf{q}_2, \ldots\}$ in real-time by sampling from a given query object whose class is unknown. It is required to estimate conditional probability for class $\mathcal{O}_i$ given $k$ samples, i.e., $P_k(\mathcal{O}_i|\mathbf{q}_1, \ldots \mathbf{q}_k)$. We propose to use either an iterative process with any trained classifier or sequential Bayesian updating. An illustration of the process is seen in Figure 2. Note that since the grasp parameterization is independent of any coordinate frame, the object is not required to remain static with respect to some world coordinate frame. Grasp of the object in any pose is sufficient and the object can move in between grasps.

## 4.1 Iterative classification

Inspired by Classification Voting Ensemble (CVE) (Li et al., 2014) and proposed by Kahanowich and Sintov (2021), we track the scores of the classes based on the predictions for each sample provided by any chosen classifier. An iterative classification (IC) process to do so is described in Algorithm 1. The algorithm maintains a vector $\mathbf{s} = (s_1, ..., s_m)$ of cumulative scores for the classes. In each iteration, a grasp is sampled and parameterized, followed by acquiring a class probability distribution $\mathbf{p}$ using function $h$. Function $h$ can be any trained classifier that outputs a class probability distribution. The highest probability $p_i \in \mathbf{p}$ is the iteration score for class $i$ and is added to $s_i$. This process is repeated until the normalized cumulative score $\hat{s}_{max} = \max(\hat{\mathbf{s}})$, where $\hat{s}_j = s_j / (\sum_i s_i) \in \hat{\mathbf{s}}$, for some class reaches above a lower bound $\lambda_s \in [0, 1]$. Normally, $\lambda_s$ will be chosen to be around $0.85 - 0.98$ for efficient performance. One may view $\hat{s}_j \in \mathbf{s}$ as the probability approximation for grasping $\mathcal{O}_j$ after some number $k$ of iterations, i.e., $\hat{s}_j \approx P_k(\mathcal{O}_j | \mathbf{q}_1, \dots \mathbf{q}_k)$.

CVE is generally used to combine the prediction scores of several classifiers, either re-trained ones or different classification types. However, in this work we use only a single classifier and combine the prediction scores over various grasp samples. In addition, only the score of the high-probability class is accumulated in each iteration. It is also possible to fully accumulate all class scores, as in CVE, by updating $\mathbf{s}$ with all iteration probabilities, i.e., replace lines 6-7 in Algorithm 1 with $\mathbf{s} \leftarrow \mathbf{s} + \mathbf{p}$. However, this will result in an excessive number of iterations for ill-trained classifiers while requiring a carefully tuned $\lambda_s$. Also, preliminary results show that for a more accurate classifier, this would provide only marginal accuracy improvement with a higher number of iterations.

```
Algorithm 1: iterative_classification($\lambda_s$)
1  Initiate elements of $\mathbf{s} = (s_1, ..., s_m)$ to 0;
2  repeat
3      Sample grasp $\mathcal{G}$;
4      $\mathbf{q} \leftarrow \Phi_n(\mathcal{G})$;
5      $\mathbf{p} \leftarrow h(\mathbf{q})$;
6      $i \leftarrow arg\max(\mathbf{p})$;
7      $s_i \leftarrow s_i + p_i$;
8      $o \leftarrow arg\max(\mathbf{s})$;
9      if first iteration then
10         $\hat{s}_{max} \leftarrow s_o$;
11     else
12         $\hat{s}_{max} \leftarrow s_o / (\sum_i s_i)$;
13 until $\hat{s}_{max} > \lambda_s$;
14 return $o$ ;            /* return class index */
```

Let $P_h(l = j | \mathcal{O}_i)$ be the probability for classifier $h$ to assign label $j$ to a grasped object $\mathcal{O}_i$ such that

$$\sum_{j=1}^{m} P_h(l = j | \mathcal{O}_i) = 1. \tag{11}$$

**Definition 1.** *Classifier $h$ is sufficient if it satisfies*

$$P_h(l = i | \mathcal{O}_i) > P_h(l = j | \mathcal{O}_i) \tag{12}$$

*for any $i, j \in \{1, ..., m\}$ and $j \neq i$.*

The above definition states that a sufficient classifier is more likely to correctly identify all objects in the set. Naturally, higher values of $P_h(l = i | \mathcal{O}_i)$ for all $i \in \{1, \dots, m\}$ mean a more accurate classifier. A classifier may not be sufficient due to limited data, incompatible classification algorithm or non-optimal NN architecture. It is worth mentioning that, in practice, the $(j, i)$ component of a classifier's confusion matrix is an approximation of $P_h(l = j | \mathcal{O}_i)$.

**Theorem 2.** *Given a sufficient classifier $h$, it must be that the expected normalized cumulative score $\mathbb{E}(\hat{s}_i | \mathcal{O}_i)$ satisfies*

$$\mathbb{E}(\hat{s}_i | \mathcal{O}_i) > \mathbb{E}(\hat{s}_j | \mathcal{O}_i), \quad \forall j \neq i, \tag{13}$$

*when using Algorithm 1.*

*Proof.* Let us assume, for now, that prediction probability of incorrect predictions tends to be lower than the prediction probability for correct ones (Hendrycks and Gimpel, 2017). Hence, given $p_{max} = \max(\mathbf{p})$, the expected value for $p_{max}$ when successfully classifying object $\mathcal{O}_i$ would be larger than an erroneous prediction. That is, statement

$$\mathbb{E}(p_{max} | l = i, \mathcal{O}_i) > \mathbb{E}(p_{max} | l = j, \mathcal{O}_i) \tag{14}$$

holds for any $j \neq i$. According to Algorithm 1, vector $\mathbf{s}$ accumulates scores for class predictions with the increase of iterations. In addition, a score is given to $s_j$ only if label $l = j$ is assigned to the query object in a particular iteration. Hence, the expected normalized value $\hat{s}_j$ of component $s_j \in \mathbf{s}$ given object $\mathcal{O}_i$ is

$$\mathbb{E}(\hat{s}_j | \mathcal{O}_i) = \mathbb{E}(p_{max} | l = j, \mathcal{O}_i) P_h(l = j | \mathcal{O}_i) \tag{15}$$

for any $j \in \{1, \dots, m\}$. From Definition 1 and (14)-(15), it must be that (13) is satisfied. Even with a more strict assumption, where $p_{max}$ for any prediction (erroneous or not) has a uniform distribution $p_{max} \sim U(\frac{1}{m}, 1)$ such that

$$\mathbb{E}(p_{max} | l = i, \mathcal{O}_i) = \mathbb{E}(p_{max} | l = j, \mathcal{O}_i) = \frac{m+2}{2m}, \tag{16}$$

statement (13) remains valid due to (12) and (15). □

The above theorem implies that as long as a classifier satisfies Definition 1 and while holding object $\mathcal{O}_i$, the cumulative score $\hat{s}_i$ will increase and converge to an higher value than $\hat{s}_j$ ($j \neq i$), with the increase of iterations, i.e., $\hat{s}_i$ is more likely to be maximal and equal to $\hat{s}_{max}$. Hence, the certainty about the prediction will grow with the addition of more samples. In turn, this will result in continuous improvement of the classifiers success rate. Let us assume that some classifier is not sufficient such that $m_p \in \{0, 1, \ldots, m\}$ is the number of grasped objects that does satisfy (12). When $m_p < m$, the classification success rate will increase only for these objects while declining for the remaining $m - m_p$ ones. This means that the success rate upper limit for a certain classifier is

$$\eta = \frac{m_p}{m} \times 100\%. \tag{17}$$

If a classifier is sufficient such that $m_p = m$, the total success rate would converge to 100% with the increase of iterations. The convergence rate depends on the quality of the classifier, that is, on the accuracy $P_h(l = i|\mathcal{O}_i)$ and whether (14) is satisfied. Hence, the proper amount of iterations to reach some level of accuracy or certainty is not known beforehand. Consequently, we cannot set the termination criterion to some arbitrary number of iterations and Algorithm 1 terminates when reaching some certainty above a threshold $\lambda_s$.

## 4.2 Sequential Bayesian Update

In an alternative approach, we consider Bayesian classification (BC) (Thrun et al., 2005). Since only rigid objects are considered, we assume a Markovian model where samples are mutually independent such that $P(\mathbf{q}_k|\mathcal{O}_t, \mathbf{q}_1, \ldots, \mathbf{q}_{k-1}) = P(\mathbf{q}_k|\mathcal{O}_t)$. The joint probability density function given $k$ samples is, therefore, expressed by the Bayes rule

$$P_k(\mathcal{O}_t|\mathbf{q}_1, \ldots \mathbf{q}_k) \propto P(\mathcal{O}_t) \prod_{i=1}^{k} P(\mathbf{q}_i|\mathcal{O}_t), \tag{18}$$

where $P(\mathcal{O}_t)$ is the prior distribution before observing the samples (Kay, 1993). Probability $P(\mathbf{q}_i|\mathcal{O}_t)$ is the likelihood of observing sample $\mathbf{q}_i$ given object $\mathcal{O}_t$. The likelihood can be approximated by using the Kernel Density Estimator (KDE) while observing the data for each object (Parzen, 1962). KDE is a non-parametric method to learn and estimate a probabilistic density function automatically from data. Given the $M_t$ independent and identically distributed training samples $\{\mathbf{q}_1^{(t)}, \ldots, \mathbf{q}_{M_t}^{(t)}\} \subset Q_M$ corresponding to object $\mathcal{O}_t$, KDE is formally defined as

$$P(\mathbf{q}_i|\mathcal{O}_t) = \frac{1}{M_t} \sum_{j=1}^{M_t} K_\sigma\left(\mathbf{q}_i - \mathbf{q}_j^{(t)}\right) \tag{19}$$

where $K_\sigma : \mathbb{R}^w \to \mathbb{R}$ is a smooth function, termed the *kernel* function, with bandwidth $\sigma > 0$. A common choice, as in this work, is a Gaussian kernel given by

$$K_\sigma(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{1}{2\sigma^2}\mathbf{x}^2\right)}. \tag{20}$$

Algorithm 2 describes the iterative process with some resemblance to Algorithm 1. For each sampled and parameterized grasp $\mathbf{q}$, the likelihood $P(\mathbf{q}|\mathcal{O}_i)$ for each object ($i = 1, \ldots, m$) is computed, followed by updating the posterior probability vector $\mathbf{p}$. This process is repeated until the posterior probability for some class reaches above a lower bound $\lambda_p \in [0, 1]$. Here also, $\lambda_p$ should be chosen to be around $0.85 - 0.98$ for efficient performance. The prior probability distribution for $\mathbf{p}_{prior} = (P(\mathcal{O}_1), \ldots, P(\mathcal{O}_m))$ can either be chosen naively (Naive Prior - NP) to be $P(\mathcal{O}_i) = \frac{1}{m}$ ($i \in \{1, \ldots, m\}$) or based on the initial prediction of some other classifier $h$ previously trained on the data (Initial Prior - IP).

**Algorithm 2:** `bayesian_classification`$(\lambda_p, \mathbf{p}_{prior})$

```
Initiate $\mathbf{p} = (p_1, ..., p_m)$ to $\mathbf{p}_{prior}$;
repeat
    Sample grasp $\mathcal{G}$;
    $\mathbf{q} \leftarrow \Phi_n(\mathcal{G})$;
    for $i = 1 \to m$ do
        Compute $P(\mathbf{q}|\mathcal{O}_i)$ using KDE;
        $p_i \leftarrow p_i \cdot P(\mathbf{q}|\mathcal{O}_i)$;
    $\mathbf{p} \leftarrow \mathbf{p}/\left(\sum_i p_i\right)$ ; /* Norm. to sum. 1 */
    $o \leftarrow arg\max(\mathbf{p})$;
until $p_o > \lambda_p$;
return $o$ ;          /* return class index */
```

## 4.3 Scale invariant grasp representation

While the grasp parameterization described in Section 3 is injective and independent of any coordinate system, it remains dependent of object size. Two objects of the same shape but with different size will have a non-equal parameterization vector resulting in a non-scalable classifier. Hence, a model trained on a set of objects will not be able to accurately classify a scaled version of the same objects. The scaling property can be useful in order to classify objects of various sizes with the same shape (e.g., balls, tubes, fruits, etc.), or avoiding to re-train a model when the size of a known object changes. A solution for this would be to normalize the grasp. Hence, we first define the scaling of an object.

**Definition 2.** *Object $\mathcal{O}_i$ is uniformly scaled by $\xi > 0$ to $\mathcal{O}_i^{(\xi)}$ if every point $\mathbf{p}_k \in \mathcal{O}_i$ is mapped to $\xi \cdot \mathbf{p}_k$, such that $\xi \cdot \mathbf{p}_k \in \mathcal{O}_i^{(\xi)}$.*

Throughout this paper, uniform scaling is referred as scaling. Next, we define the scaling of a grasp.

**Definition 3.** *Let $\mathcal{G}_i$ be a grasp of object $\mathcal{O}_i$ at contact points $\mathcal{P}_i$. A scaled grasp $\mathcal{G}_i^{(\xi)}$ corresponds to grasping* *of object* $\mathcal{O}_i^{(\xi)}$ *at contact points* $\xi \cdot \mathcal{P}_i$.

The above definition addresses the scaling of the grasp polyhedron $\Theta(\mathcal{P}_i)$ by scaling the vertices by $\xi$. The direction of the normals at the contact points do not change, i.e., scaling of grasp $\mathcal{G} = \{\mathcal{P}, \mathcal{N}\}$ by $\xi$ results in $\mathcal{G}^{(\xi)} = \{\xi \cdot \mathcal{P}, \mathcal{N}\}$. We note that scaling of a grasp can also be accompanied with rotation and translation of $\mathcal{G}$ while not affecting the parameterization as described in Theorem 1. Without loss of generality, we omit possible transformation when scaling a grasp for simplicity of presentation in the remaining part of the paper.

Let $a_{vu}$ be the surface area of facet $u$ on polyhedron $\Theta(\mathcal{P}_v)$ ($u = 1, ..., U$ where $U$ is the number of facets of $\Theta(\mathcal{P}_v)$), we define $A_v = \sqrt{\sum_{u=1}^{U} a_{vu}}$ to be the square root of the total surface area of polyhedron $\Theta(\mathcal{P}_v)$. We now argue that the scaling of two grasps corresponding to scaled objects, each by its own squared total surface area, will result in a scale invariant grasp representation.

**Theorem 3.** *Given grasps* $\mathcal{G}_i$ *and* $\mathcal{G}_j$ *of objects* $\mathcal{O}_i$ *and* $\mathcal{O}_j$*, respectively, where some unknown scaling factor* $\xi$ *exists such that* $\mathcal{O}_j \equiv \mathcal{O}_i^{(\xi)}$ *and* $\mathcal{G}_j = \mathcal{G}_i^{(\xi)}$. *By using scaling factors* $A_i^{-1}$ *and* $A_j^{-1}$*, it must be that*

$$\Phi_n\left(\mathcal{G}_i^{\left(A_i^{-1}\right)}\right) = \Phi_n\left(\mathcal{G}_j^{\left(A_j^{-1}\right)}\right) \tag{21}$$

*for any number of fingers* $n$.

*Proof.* See Appendix B. □

Theorem 3 states that a scale invariant training set can be generated by scaling each grasp of an object with the total surface area of the formed polyhedron. Then, a trained classifier can identify the scaled objects corresponding to the original ones. The scale invariance property could remove the need for recollection and retraining of a classifier for a new object that is a scaled version of an already known one. In addition, scaling can enable generalization for geometry classification as will be shown in the results.

## 4.4 Embedded $z$-finger grasps

While sampling an $n$-finger grasp, one may consider it as a set of possible $z$-finger grasps, where $z < n$. In an $n$-finger grasp, there are $c_{n,z}$ combinations of $z$-finger grasps given by

$$c_{n,z} = \binom{n}{z} = \frac{n!}{z!(n-z)!}. \tag{22}$$

Therefore, we may use a classifier trained over $z$-finger grasps to better exploit the information given by a single $n$-finger sample. Hence, potentially reduce the overall number of physical grasp samples. Algorithms 3 and 4 present the modifications of the IC and BC algorithms, respectively, to use $z$-finger grasps. For every grasp sample $\mathcal{G}$, all possible combinations of $z$-finger grasps $\mathcal{G}_j^z$ ($j \in \{1, ..., c_{n,z}\}$) are used to update classification certainty as described in Sections 4.1 and 4.2.

**Algorithm 3:** `iterative_z_classification`($\lambda_s$)

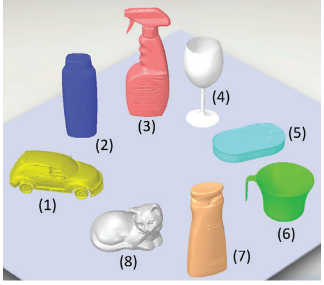
```
Initiate elements of $\mathbf{s} = (s_1, ..., s_m)$ to 0;
while $True$ do
    Sample grasp $\mathcal{G}$;
    $\{\mathcal{G}_1^z, \ldots, \mathcal{G}_{c_{n,z}}^z\} \leftarrow$ all $z$-finger combinations in $\mathcal{G}^n$;
    for $j = 1 \rightarrow c_{n,z}$ do
        $\mathbf{q} \leftarrow \Phi_z(\mathcal{G}_j^z)$;
        $\mathbf{p} \leftarrow h(\mathbf{q})$;
        $i \leftarrow arg\max(\mathbf{p})$;
        $s_i \leftarrow s_i + p_i$;
        $o \leftarrow arg\max(\mathbf{s})$;
        if first iteration then
            $\hat{s}_{max} \leftarrow s_o$;
        else
            $\hat{s}_{max} \leftarrow s_o / (\sum_i s_i)$;
        if $\hat{s}_{max} > \lambda_s$ then
            return $o$ ; /* ret. class index */
```

**Algorithm 4:** `bayesian_z_classification`($\lambda_p, \mathbf{p}_{prior}$)

```
Initiate $\mathbf{p} = (p_1, ..., p_m)$ to $\mathbf{p}_{prior}$;
while $True$ do
    Sample grasp $\mathcal{G}^n$;
    $\{\mathcal{G}_1^z, \ldots, \mathcal{G}_{c_{n,z}}^z\} \leftarrow$ all $z$-finger combinations in $\mathcal{G}^n$;
    for $j = 1 \rightarrow c_{n,z}$ do
        $\mathbf{q} \leftarrow \Phi_z(\mathcal{G}_j^z)$;
        for $i = 1 \rightarrow m$ do
            Compute $P(\mathbf{q}|\mathcal{O}_i)$ using KDE;
            $p_i \leftarrow p_i \cdot P(\mathbf{q}|\mathcal{O}_i)$;
        $\mathbf{p} \leftarrow \mathbf{p} / (\sum_i p_i)$;
        $o \leftarrow arg\max(\mathbf{p})$;
        if $p_o > \lambda_p$ then
            return $o$ ; /* ret. class index */
```

Considering a lower number of fingers than the hand has can also be useful in exploiting incomplete grasps. An $n$-finger robotic hand may not always be able to make contact of all fingers with the object. Feedback for lack of finger contact can be acquired from joint torques or some low-level haptics, if available. In such case, the $z < n$ fingers that achieved contact can also provide viable information without the need of a new grasp.

Hence, classification using either IC or BC can be done with a sampling sequence of an heterogeneous number of fingers, e.g., grasp sequence $\{\mathcal{G}^n, \mathcal{G}^{z_1}, \mathcal{G}^{z_2}, \mathcal{G}^n, \mathcal{G}^{z_3}, \dots\}$ where $z_j < n$ may be acquired. Each $\mathcal{G}^j$ is parameterized and classified based on the corresponding trained model.



| | Object |
|---|---|
| (1) | ToyCarYelloq |
| (2) | Shampoo |
| (3) | SprayFlask |
| (4) | Wineglass |
| (5) | HeringTin |
| (6) | GreenCup |
| (7) | ShowerGel |
| (8) | CatLying |

Figure 5: Eight objects from the KIT object models database used in the experiments.

# 5 Experimental Results and Analysis

In this section, we test and analyse the proposed method over a set of distinct objects while not considering the kinematics of the robotic hand. Experiments with robotic hands are presented in Section 6. We have picked eight objects from the KIT object models database (Kasper et al., 2012) shown in Figure 5. The KIT database is a set of objects scanned with a 3D digitizer, resulting in a triangular mesh of 800 faces for each object. The eight objects were selected because they offer a wide diversity of shapes while some have irregular ones (the *ToyCarYelloq* and the *CatLying*), and others have relatively similar shapes that are challenging to make a distinction (*Shampoo* and *ShowerGel*). We first present the process of data collection and training of various classifiers over the set of CAD models. In Section 5.2, we present results for object recognition using IC and BC. Further, we test the scalability property and the use of $z$-finger grasps. Finally, we demonstrate the ability to classify types of geometries over a larger set of objects. Code and data-sets are publicly available in a dedicated Git repository[1].

## 5.1 Data collection and model training

The KIT objects are represented as triangular meshes. In the context of possible contact locations, each mesh is converted to a set $\mathcal{F}_l$ of contact locations (centers of the triangles) and the normals at the contacts (normals to the triangles) directed inwards. We generate a training set as described in Section 4. An $n$-finger grasp is sampled by randomly selecting $n$ elements from $\mathcal{F}_l$ (with or without normals). A training set is acquired by sampling and parameterizing $N = 200,000$ $n$-finger grasps for each object, and labeling them. We note that 15% of the training data was used for validation during training. The testing phase, from which we present results, was performed after training and done by directly sampling grasps from the objects. The average generation time of data for each object is 4.72 minutes computed on an Intel-Core i7-9700 Ubuntu machine with 16 GB of RAM.

For IC and for the prior of BC-IP, we have trained and tested different classifiers, including: Nearest-Neighbors, Support Vector Machines (SVM), Gaussian Processes, Random Forests, Neural-Network and Decision Trees. The SVM classifier was trained using linear and Radial Basis Function (RBF) kernels. The NN architectures were formed and optimized individually for different input dimensions, depending on the number of fingers and whether the normals were included. For example, a 4-finger grasp with normals, that yields an input dimension of 14, used a deep NN of seven hidden layers of 500 width. A 9-finger grasp with normals and parameterization vector of dimension 39 reached the best results with a NN of eight layers and width of 289 neurons. The networks all used a Rectified Linear Unit (ReLU) activation function and were trained with the back-propagation algorithm.

Table 1 reports the classification success rate of 1,000 trials per object with 3-, 4- and 5-finger grasps directly sampled from the objects. We note that these are the rates for a one-shot without any iterative classification, i.e., classification based on a single grasp sample. Most of the classification techniques performed poorly, while the NN outperforms. However, it is clear that a single grasp sample generally cannot be used to reliably classify an object. As discussed in Section 4, one grasp and its parameterization does not embed the entire information of the object and is not sufficient. Hence, additional grasps can be used to increase the certainty about the object. We use the NN classifier in IC and a prior for BC-IP in further experiments. Results for iterative object recognition are presented next.

## 5.2 Object Recognition

In this section, we experimentally show and analyse the success rate and the required number of samples to classify an object through contact locations, with and without including the normals at the contacts. For both IC and BC, we have tested ranges of values for $\lambda_s$ and $\lambda_p$. The value $\lambda_s = \lambda_p = 0.85$ has yielded sufficient results,

[1] https://github.com/avishais/Grasp_Classification

Table 1: Success rate comparison for different classifiers

| Num. fingers |  | 3 |  | 4 |  | 5 |  |
|---|---|---|---|---|---|---|---|
| normals |  | w/ | w/o | w/ | w/o | w/ | w/o |
| Nearest Neighbors | % | 70.8 | 45.6 | 59.1 | 56.1 | 62.2 | 61.8 |
| Linear SVM |  | 41.2 | 41.0 | 50.0 | 49.6 | 55.6 | 55.7 |
| RBF SVM |  | 76.9 | 46.9 | 47.6 | 57.8 | 37.3 | 63.5 |
| Gaussian Process |  | 69.5 | 42.7 | 59.0 | 57.3 | 64.7 | 63.8 |
| Random Forests |  | 66.8 | 47.3 | 58.7 | 57.7 | 64.6 | 63.3 |
| Neural Network |  | 86.0 | 47.0 | 80.7 | 62.4 | 83.7 | 71.1 |
| Decision Trees |  | 61.9 | 45.7 | 58.2 | 54.4 | 61.8 | 61.0 |

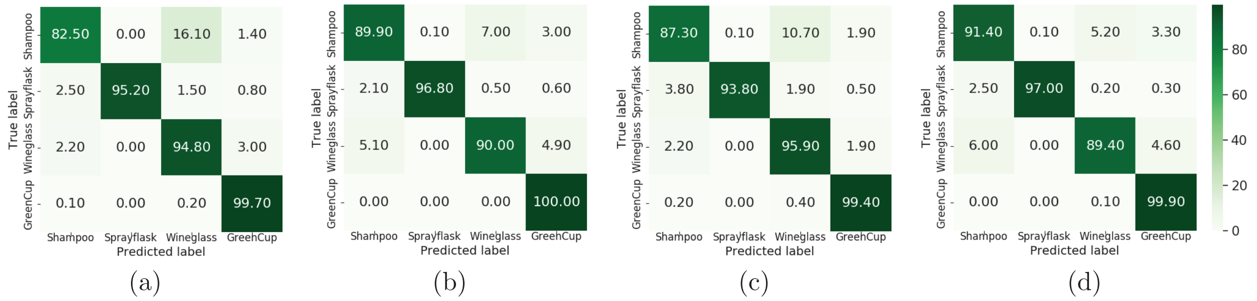


Figure 6: Confusion matrices for classification of four objects using BC-NP. The results are for 1,000 query trials per object. Classification with a 4-finger grasp (a) with and (b) without the normals at the contact points, and a 5-finger grasp (c) with and (d) without the normals. Average number of re-sampling iterations is (a) 4.87, (b) 6.52, (c) 3.95 and (d) 5.21, for $\lambda_p = 0.85$.

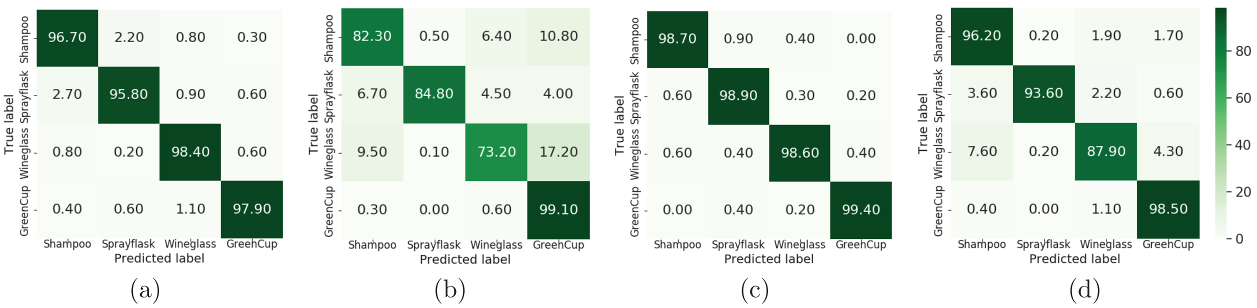
Table 2: Baseline success rate results for object recognition using 12 grasp samples

| Num. fingers | 3 |  | 4 |  | 5 |  |
|---|---|---|---|---|---|---|
| normals | w/ | w/o | w/ | w/o | w/ | w/o |
| ICP (%) | 23.6 | 17.2 | 27.1 | 20.9 | 41.2 | 27.5 |

both in terms of success rate and minimum number of grasp samples. Surely this number can be higher to improve the success rate at the cost of more grasp samples. We use this value in the following experiments while later showing an analysis of their effect on performance and also the success rate with regards to the number of grasp samples.

For a clear presentation, we first show results for classifying the four object: *Shampoo*, *SprayFlask*, *Wineglass* and *GreenCup*. A baseline is set for fair comparison where a more naive search is included. Here, contact point locations without any parameterization are directly searched on the surface of the CAD model using the Iterative Closest Point (ICP) algorithm (Besl and McKay (1992)) and without any learning. In ICP, the grasp points are matched to the mesh points of the CAD models of all objects to find the lowest residual. Table 2 shows the results for classification with ICP while including 12 grasps sampled from the surface of the CAD model. Direct naive search on the surface of the objects provides poor results. The points from the hand are very sparse and not sufficient for direct point cloud matching. Much more grasp samples are required in order to provide sufficient performance. The baseline results motivate a learning approach.

Figure 6 shows confusion matrices for classification with BC-NP for 4- and 5-finger grasps with and without including the normals. Not using the normals requires an average of about one more sample in order to reach the lower bound probability. Similarly, Figure 7 shows results for classification with IC. Here, not using the normals required less iterations with the cost of a lower success rate. Complete results for 3 to 10 finger grasps with the eight objects are shown in Figures 8 and 9, with and without including the normals, respectively. The figures report results for the classification success rate and the required number of samples. Table 3 shows the average number of iterations including corresponding standard-deviations. When not considering the normals, 2-finger grasps are included as discussed in Section 3. Generally, all three methods (i.e., IC, BC-NP and BC-IP) acquire high success rate with a slight advantage for IC. However, IC performs poorly when not including normals and with a low number of fingers (2-3), as seen in Figure 9. NN-based classifiers trained



(a) (b) (c) (d)

Figure 7: Confusion matrices for classification of four objects using IC. The results are for 1,000 query trials per object. Classification with a 4-finger grasp (a) with and (b) without the normals at the contact points, and a 5-finger grasp (c) with and (d) without the normals. Average number of re-sampling iterations is (a) 4.35, (b) 3.6, (c) 4.67 and (d) 3.97, for $\lambda_p = 0.85$.

over grasps with a low number of fingers and no normals failed to satisfy condition (12). Hence, they exhibited poor results for IC. In addition, the use of two fingers produced relatively high success rate with BC, but with the high cost of too many grasp samples. In conclusion, IC is preferred for object recognition with normals as it can better embed high dimensional data. Similarly, BC provides better results for object recognition without normals utilizing the strength of KDE for lower dimensional data. Overall, using more fingers provides more information that decreases the need for additional grasp samples. This motivates the use of $z$-finger grasps to be tested in the next section.

When comparing between BC-NP and BC-IP, the former reaches better success rate while the latter requires less samples. While the BC-IP requires less samples in cases where the prior for a certain object is high enough, the prior can also wrongly bias the classification when the probabilities for all objects are rather low. Also, the accuracy of BC-NP decreases, when considering normals and with the addition of more fingers, due to the difficulty to cope with the increase of dimensionality. In general, IC and BC perform better in queries with and without normals, respectively.

Figure 15 presents the success rate for 4-finger grasps with regards to the number of grasps samples required. The results shown in the figure will be addressed in more detail in the next section. Similar behavior can be seen when observing the performance of IC and BC with regards to the choice of $\lambda_s$ and $\lambda_p$, respectively. In the above evaluations, the value $\lambda_s = \lambda_p = 0.85$ was chosen as a compromise between success rate and number of samples. Figure 12 shows the success rate and mean number of samples with respect to the choice of $\lambda_s$ and $\lambda_p$ for 4-finger grasps. Results show that, generally, more grasps samples are required in order to improve the success rate.

Figure 10 depicts recognition success rate with BC-NP, with regards to the number of classified objects and for 5-finger grasps. The results include up to 50 objects from the KIT dataset (including the eigh t previously used). In addition, cross validation was performed over different sets of objects. As stated previously, BC is better when not including normals and, therefore, success rate declines moderately with the addition of objects. Nevertheless, the results demonstrate the ability to classify a large number of objects. Figure 13 presents the confusion matrix of object recognition for 5-finger grasps without including the normals. The results show that accurate object recognition can be performed for a large object set without including normals.

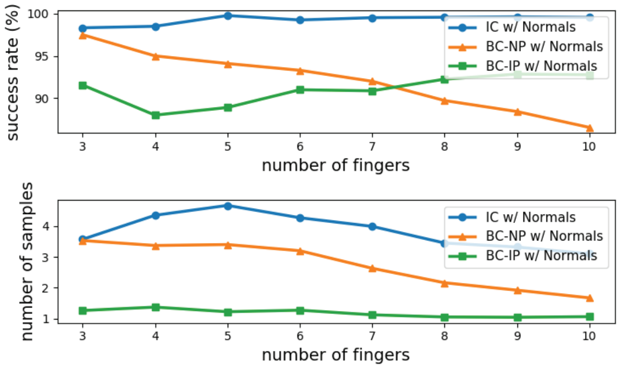


Figure 8: Success rate (top) and average number of grasp samples (bottom) with regards to the number of fingers used in the sampling while using the normals at the contacts. Plots show results for IC, BC-NP and BC-IP.

Parameterization uncertainties can occur due to noise in finger joint feedback and affect the accuracy of $\mathcal{P}$. Therefore, we analyze the impact of zero mean Gaussian noise in $\mathcal{P}$ on recognition success rate. Figure 11 presents results for the success rate with regards to the

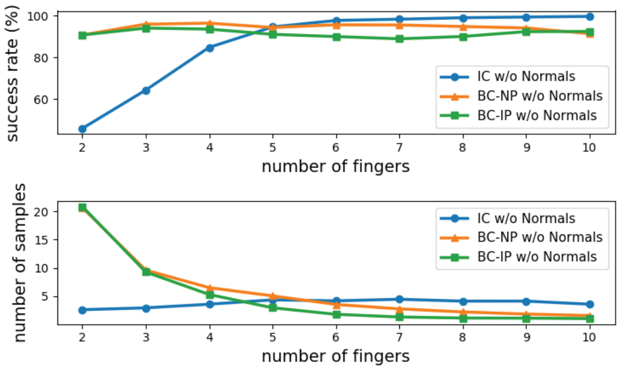


Figure 9: Success rate (top) and average number of grasp samples (bottom) with regards to the number of fingers used in the sampling while **not** using the normals at the contacts. Plots show results for IC, BC-NP and BC-IP.

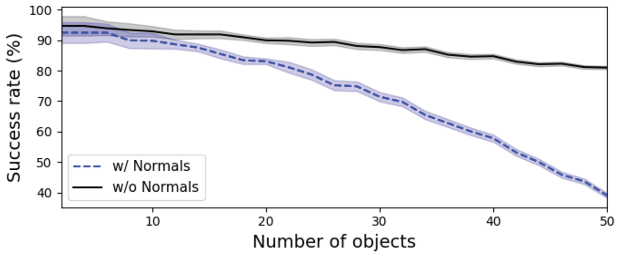


Figure 10: Success rate with regards to the number of classified objects for BC-NP with 5-finger grasps and without using the normals at the contacts. The shadowing illustrates the standard deviation over cross validation of different object sets.

standard deviation of the noise. Such standard deviation can reach, in average, up to 3% of object size. In addition, results are for 4, 8 and 20 object sets with cross validation over different objects. Results show slight decrease in performance while still able to provide a high recognition success rate with noisy signals.

We next observe performance of the methods with regards to size of the training data. Results for success rate and average number of iterations with regards to the number of training points can be seen in Figure 14. The results show that less than half of the data is required to achieve sufficient accuracy. Nevertheless, the data is collected relatively fast (a few minutes) through a computation process without the need for sampling from the physical objects. Hence, collecting excessive data is relatively cheap. In addition, the results show the benefit of improving accuracy using IC when comparing between one-shot classification accuracy (dotted blue curves) and IC (solid blue curves).

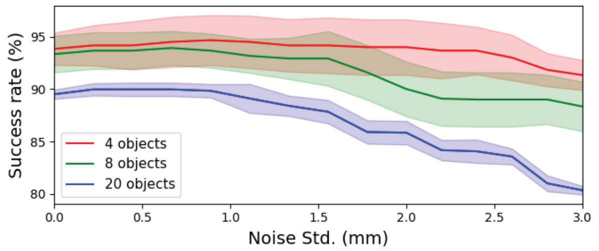


Figure 11: Success rate with regards to the standard deviation of Gaussian noise added to the set of contact points $\mathcal{P}$, with zero mean. Results are for BC-NP with 5-finger grasps and without using the normals at the contacts. The shadowing illustrates the standard deviation over cross validation of different object sets.

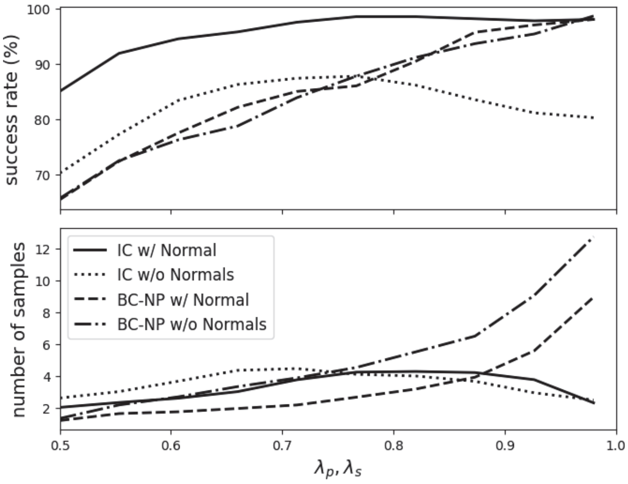


Figure 12: Success rate (top) and average number of grasp samples (bottom) with regards to $\lambda_s$ and $\lambda_p$ for IC and BC-NP, respectively, and for 4-finger grasps.

## 5.3 $z$-finger grasps

In this Section, we evaluate performance improvement when physically sampling $n$-finger grasps and use classifiers trained over $z$-finger grasps ($z < n$), i.e., using algorithms 3 or 4. Experiments were performed as previously discussed while using classifiers (either KDE for BC or NN for IC) trained over 3- and 4-finger grasps ($z = 3$ or $z = 4$, respectively). For $z = 3$, we test $n = \{4, 5, 6, 7\}$ finger grasps, having $c_{n,3} = \{4, 10, 20, 35\}$ combinations of 3-finger grasps, respectively. Similarly, for $z = 4$, we test $n = \{5, 6, 7\}$ finger grasps corresponding to $c_{n,4} = \{5, 15, 35\}$ grasp combinations, respectively. In practice, we randomly sample four combinations out of the $c_{n,z}$ as preliminary tests show that more do not provide significantly better results. Table 4 presents the success rate and the

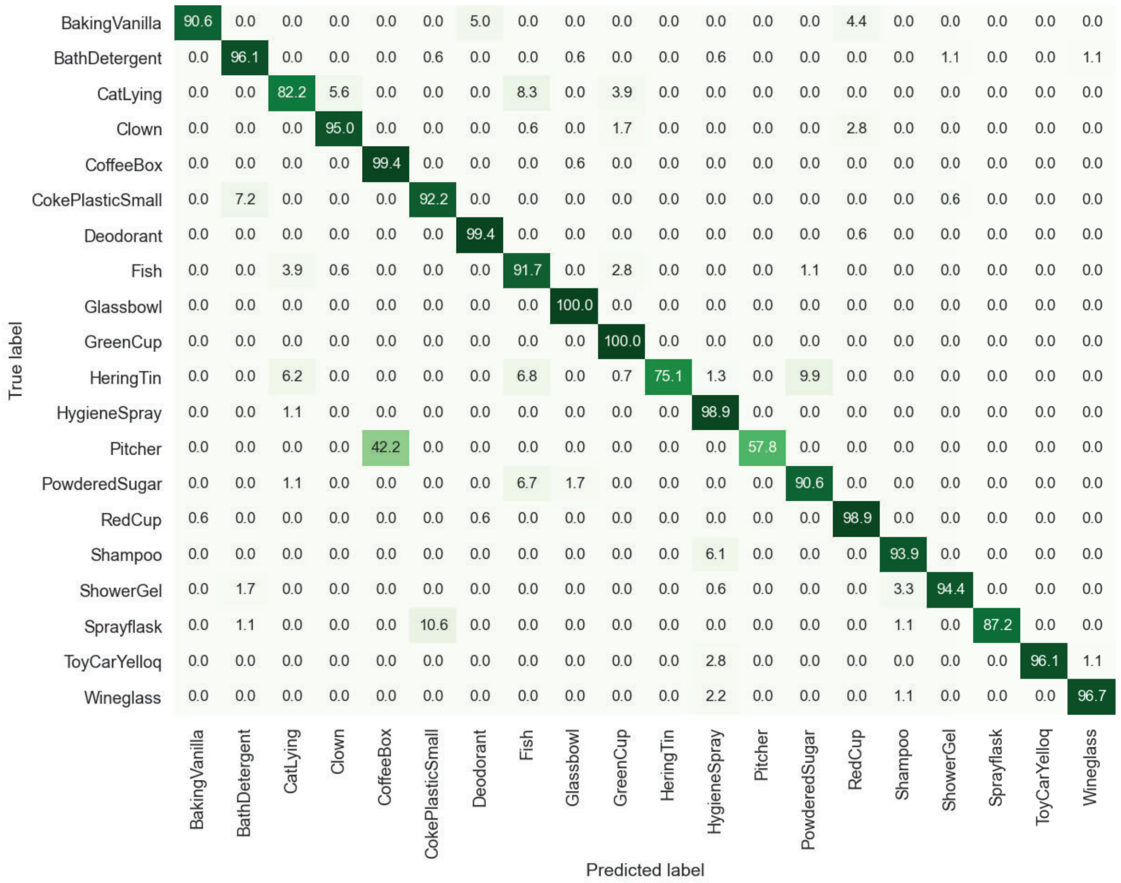


Figure 13: Confusion matrix for classification of 20 objects using BC-NP with 5-finger grasps and without normals.

Table 3: Number (mean and std.) of grasp samples

| # fingers | 3 | | 4 | | 5 | |
|---|---|---|---|---|---|---|
| normals | w/ | w/o | w/ | w/o | w/ | w/o |
| | IC | | | | | |
| Mean | 3.57 | 2.96 | 4.35 | 3.60 | 4.67 | 4.37 |
| Std. | 4.57 | 3.25 | 5.80 | 4.45 | 6.43 | 5.70 |
| | BC-NP | | | | | |
| Mean | 3.53 | 9.62 | 3.37 | 6.52 | 3.40 | 5.07 |
| Std. | 2.07 | 13.81 | 1.90 | 4.87 | 1.46 | 3.04 |

average number of $n$-finger grasp samples taken when using IC and BC-NP. The table also presents the reduction in the average number of samples compared to the results for regular sampling presented in Section 5.2. In all methods, the success rates when using the normals are similar to previous ones (Figure 8) while the average number of samples is significantly reduced by 40%-70%. However, when not using the normals and for IC, reduction exists while the success rates are low. This is since the original NN classifiers for $n = 3$ and $n = 4$ performed poorly as seen in Table 1 and Figure 9. Nevertheless, BC-NP yielded better success rates with some improvements in the required average number of samples in most cases. Conclusively, IC and BC with $z$-finger grasps provide best results for object recognition with and without normals, respectively.

Figure 15 presents classification success rates with regards to the number of grasp samples and for all the above methods. The results were generated while disregarding the bounds $\lambda_s$ and $\lambda_p$. Thus, we measured the success rate when a certain number of grasp samples is taken. Almost all methods are able to reach 100% success rate with the increase of grasp samples. However, some require an excessive amount of samples which is not practical. Nevertheless, a small number of samples is required to achieve relatively high success rate. Evidently, the use of a $z$-finger model reduces the number of samples significantly when using the normals and slightly when not using normals. We note that IC with $z = 3$ without normals converged to some success rate lower than 100% since its NN classifier does not

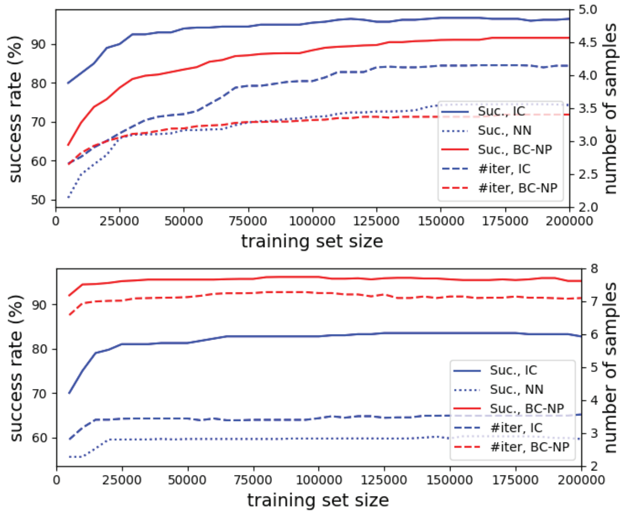


Figure 14: Classification success rate (solid curves) and average number of iterations (dashed curves) with regards to the size of the training data for IC (blue) and BC-NP (red) while using (top) and not using (bottom) the normals at the contacts. Dotted curves are the success rates for one-shot classification of the NN classifiers prior to using IC.

satisfy condition (12) for all objects.

As described in Section 4.4, $z$-finger grasps can also be used to exploit incomplete grasps when not all fingers are in contact. We test the performance of such case for 4- and 5-finger robotic hands. For each grasp instant, we assume that the number of fingers that may be in contact are given by the discrete probability distribution $p_n(z) = Pr(z = a)$. For $n = 4$ and where $a = \{3, 4\}$, the probability distribution is defined as

$$p_4(z) = \left\{ \begin{array}{lcl} Pr(z=3) & = & 0.4 \\ Pr(z=4) & = & 0.6 \end{array} \right. . \tag{23}$$

Similarly, when $n = 5$ and $a = \{3, 4, 5\}$, we define the probability distribution as

$$p_5(z) = \left\{ \begin{array}{lcl} Pr(z=3) & = & 0.2 \\ Pr(z=4) & = & 0.3 \\ Pr(z=5) & = & 0.5 \end{array} \right. . \tag{24}$$

The above probabilities were chosen arbitrary for demonstration. Grasps of less than three fingers are not taken in account and are re-sampled. Table 5 presents classification results with the above probability distributions for IC and BC-NP. The results indicate that accuracy is yet maintained even if incomplete grasps occur frequently.

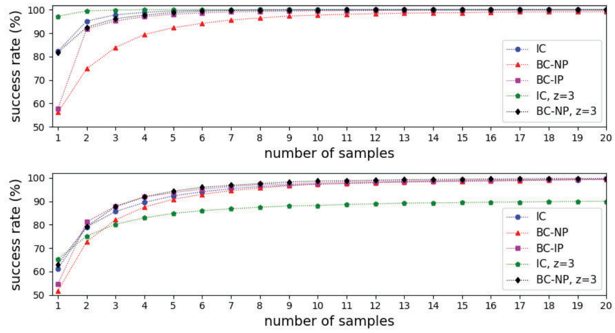


Figure 15: Classification success rate of the eight objects for 4-finger grasps (top) with and (bottom) without using the normals at the contacts.

## 5.4 Object scalability

To test the ability to classify objects of different scale, we train the classifiers as described in Section 5.2 while normalizing the grasp parameterization vectors. As described in Section 4.3, two similar grasps with a scaling factor reference will have the same parameterization vector if the contact points of each grasp are scaled by the square root of the total surface area ($A_v$). Hence, we build a training set for the eight KIT objects comprising of labeled and normalized parameterization vectors of sampled grasps. Evaluation was then performed for each object over 1,000 trials. In each trial, the object was randomly scaled by $s \in [0.1, 5]$ followed by grasp sampling and prediction using IC and BC. Table 6 summarizes the success rates for 3-, 4- and 5-finger grasps. The success rates are slightly lower compared to non-scaled objects. IC still performs poorly without normals and when using a low number of fingers. It is important to note that scaling down objects may hinder the ability to acquire valid normal vectors since the robot hand does not also scale along. Nevertheless, the results show that the classification can be scale invariant and generalized to objects of different size even without normal information.

## 5.5 Geometry recognition

In this section, we evaluate the ability of the approach to classify everyday objects into types of geometries and not only specific objects. First, we train a classifier using three selected primitive geometries: a box, a sphere and a cylinder. To generate a training set for the classifier, three spatial CAD models were formed in an arbitrary size and positioned aligned to the primary axes as seen in Figure 16. Their general size is not of an importance as further discussed. Then, for each object, 1,000 variations were created by non-uniform resizing along

Table 4: Success rate and sampling improvement when using $z$-finger grasps

| $n$ | 4 | | 5 | | 6 | | 7 | |
|---|---|---|---|---|---|---|---|---|
| Normals | w/ | w/o | w/ | w/o | w/ | w/o | w/ | w/o |
| IC | IC, $z = 3$ | | | | | | | |
| Success rate (%) | 95.7 | 62 | 96.7 | 62.8 | 96.1 | 64 | 96.8 | 62.3 |
| Num. samples | 1.6 | 1.26 | 1.57 | 1.25 | 1.57 | 1.26 | 1.49 | 1.28 |
| Samp. redu. (%) | 63.2 | 65 | 66.3 | 71.4 | 60.4 | 69.9 | 62.6 | 71.4 |
| IC | IC, $z = 4$ | | | | | | | |
| Success rate (%) | - | - | 93.5 | 72.0 | 95.0 | 76.2 | 97.0 | 76.7 |
| Num. samples | - | - | 1.75 | 1.58 | 2.2 | 1.26 | 2.2 | 1.43 |
| Samp. redu. (%) | - | - | 62.5 | 63.8 | 44.1 | 69.9 | 45.3 | 68.1 |
| BC-NP | BC-NP. $z = 3$ | | | | | | | |
| Success rate (%) | 92.5 | 91.9 | 91.7 | 93.1 | 93.4 | 96.2 | 94.5 | 96.1 |
| Num. samples | 1.33 | 4.2 | 1.27 | 4.65 | 1.32 | 4.8 | 1.3 | 5.2 |
| Samp. redu. (%) | 60.8 | 35.7 | 60.3 | 8.3 | 49.8 | -35 | 40.2 | -86 |
| BC-NP | BC-NP. $z = 4$ | | | | | | | |
| Success rate (%) | - | - | 96.0 | 89.0 | 95.5 | 91.0 | 97.2 | 94.0 |
| Num. samples | - | - | 1.36 | 2.48 | 1.29 | 2.53 | 1.31 | 2.15 |
| Samp. redu. (%) | - | - | 60 | 51.1 | 59.6 | 28.9 | 50.2 | 22.6 |

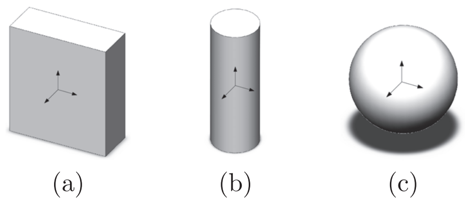


Figure 16: Three primitive geometries used to train a classifier to classify types of objects: (a) box, (b) cylinder and (c) sphere.

the primary axes with a random factor of $s_k \in [0.5, 1.5]$, where $k = x, y, z$. Thus, we acquire three sets associated with the primitive geometries. Due to the non-uniform resizing of the sphere and cylinder, their corresponding sets include ellipsoids and elliptic cylinders of various sizes, respectively. Since the primitive objects and future query ones have arbitrary sizes, grasp samples are required to be normalized in order to compare similar geometries with different sizes. Therefore, each sampled grasp is scaled by the surface area of its grasp polyhedron as described in Section 4.3. According to Theorem 3, such scaling will enable invariant grasp representation and allow to match between grasps of objects with different sizes. Training data was acquired by sampling, normalizing, parameterizing and labeling random grasps within the three sets.

We now explore the performance of IC and BC-NP, trained with the primitive data, in classifying the geometry of 33 everyday objects taken from the Graspnet dataset (Fang et al., 2020). Similar to the experiments in Section 5.2 and according to Algorithms 1-2, a set of

Table 5: Success rate and number of samples for incomplete grasps

| $n$, $p_n(z)$ | 4, $p_4(z)$ | | 5, $p_5(z)$ | |
|---|---|---|---|---|
| Normals | w/ | w/o | w/ | w/o |
| | IC | | | |
| Success rate (%) | 98.7 | 75.8 | 98.8 | 87.1 |
| Num. samples | 4.99 | 3.47 | 4.19 | 3.56 |
| | BC-NP | | | |
| Success rate (%) | 97.3 | 95.3 | 93.3 | 96.0 |
| Num. samples | 3.27 | 7.72 | 2.87 | 6.3 |

Table 6: Success rate (%) for classifying scaled objects

| $n$ | 3 | | 4 | | 5 | |
|---|---|---|---|---|---|---|
| Normals | w/ | w/o | w/ | w/o | w/ | w/o |
| IC | 96.2 | 43.3 | 90.2 | 63.8 | 91.1 | 73.5 |
| BC-NP | 91.5 | 84.0 | 92.2 | 91.3 | 85.2 | 90.9 |
| BC-IP | 93.25 | 86.8 | 91.0 | 90.5 | 90.2 | 91.8 |

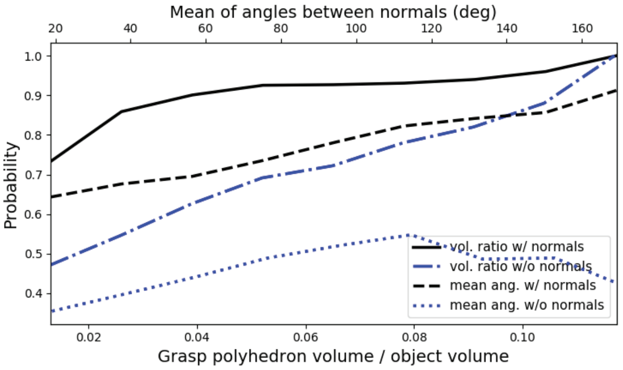


Figure 17: The average probability of a NN classification with regards to the volume ratio between the grasp polyhedron and the object, and the mean angles of between the normals at the contacts. The mean of the angles is computed using the mean of circular quantities.

4-finger grasp samples are taken from each object until the lower bound certainty ($\lambda_s = \lambda_p = 0.85$) is achieved. We note that grasp samples taken from each query object are also normalized according to Theorem 3. Table 7 presents, for each object, classification rate into each of the three primitives out of 10 trials. The results do not consider the normals at the contacts to demonstrate classification with minimal information. Experiments with normals have not shown significant advantage in this case. The results show that most of the objects are classified as expected with high success rate. IC is slightly more challenged in identifying boxes than BC-NP. Some objects, such as *dabao_wash_soup*, *soap* and *dove*, have flat faces along with large rounded corners that can be classified differently based on the samples taken. Hence, their classification rates are distributed between the different primitives. Nevertheless, our proposed approach have been successfully demonstrated to distinguish between different geometries of everyday objects.

### 5.6 Grasp Properties

We characterize the properties of a grasp in terms of its ability to provide information about a query object. Hence, we analyse the classification certainty that a grasp provides with regards to the properties of its parameterization. We consider two measures. First, we measure the ratio between the volume of the polyhedron formed by the contact points and the volume of the object. Second, we measure the mean angles between all normals at the contact points computed by the mean of circular quantities. The volume and mean angles measure the distribution of the contacts and the variation of normal directions, respectively.

Figure 17 presents results for NN classification certainty with regards to these measures for 4-finger grasps with and without normals. The certainty is expressed in the probability distribution assigned to the query object by the classifier. It is clear that increasing the grasp polyhedron's volume yields better prediction certainty. A higher volume means better sampling of shape and size and a more informative classifier input. Similarly, when considering normals, a high variation in the normal directions implies about the object shape. When not considering normals in the classifier, they do not provide additional information as expected and, therefore, the classification probability is low. These insights are similar to human behaviour where tactile recognition of an object is usually performed with a caging formation of the hand. On the other hand, positioning all of the fingers on one facet of a box, for example, provides deficient information.

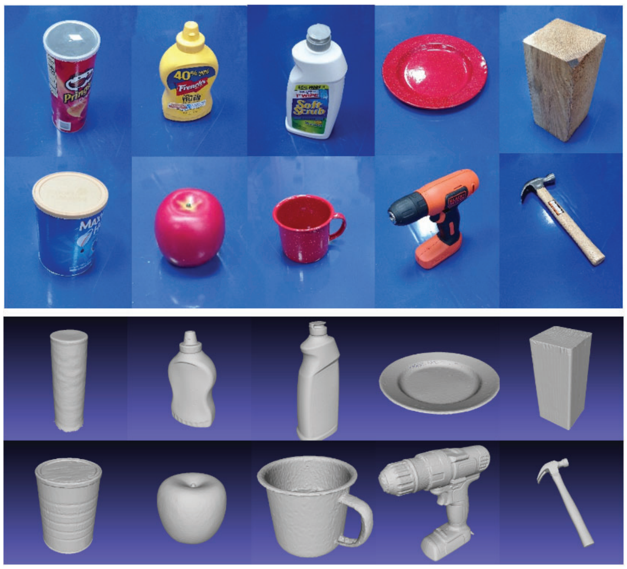


Figure 18: Ten YCB objects used in the grasping experiments (top) and their meshes (bottom) used for training the classifiers. From top left: chips can (01), mustard bottle (06), bleach cleanser (21), plate (29), wood block (36), master chef can (02), apple (13), mug (25), power drill (35) and hammer (48) (Calli et al., 2017).

## 6 Gripper Experiments

In this section, we conduct experiments while considering real grasps with robotic hands. In all experiments, we evaluate recognition capabilities with the Yale-CMU-Berkeley (YCB) dataset (Calli et al., 2017). The YCB dataset includes models of various objects specifically designed for benchmarking in grasping and manipulation research. We use 10 objects seen in Figure 18: chips can (YCB object number 01), master chef can (02), mustard bottle (06), apple (13), bleach cleanser (21), mug (25), plate (29), power drill (35), wood block (36) and hammer (48).

We experiment with a simulated 5-finger Pisa/IIT SoftHand and a real 4-finger Allegro hand. The Pisa/IIT SoftHand (Catalano et al., 2014) was simulated in ROS-Gazebo environment. The Pisa/IIT hand is an anthropomorphic hand with 19 joints and adaptive synergy capabilities. Hence, it uses only one actuator and a system of ligaments and tendons to enable adaptive grasping. In the simulation, seen in Figure 19, we assume that finger joint angles can be measured and the kinematics are known. That is, the locations of the finger tips can be computed at any time. Joint torques and, therefore, normal directions cannot be measured. In real experiments, we used the 4-finger Allegro robotic hand. The Allegro is a fully-actuated hand comprised

Table 7: Results for classifying objects from the Graspnet (Fang et al., 2020) dataset into three types of geometries using 4-finger grasps and without considering normals at the contacts

| id | Image | Name | IC (rate in %) | | | BC-NP (rate in %) | | |
|---|---|---|---|---|---|---|---|---|
| | | | box | el. cylinder | ellipsoid | box | el. cylinder | ellipsoid |
| 000 | | 003_cracker_box | 50 | 0 | 50 | 90 | 10 | 0 |
| 001 | | 004_sugar_box | 60 | 10 | 30 | 80 | 20 | 0 |
| 003 | | 006_mustard_bottle | 30 | 50 | 20 | 40 | 60 | 0 |
| 005 | | 011_banana | 0 | 100 | 0 | 0 | 100 | 0 |
| 010 | | 001_chips_can | 10 | 90 | 10 | 0 | 100 | 0 |
| 011 | | 012_strawberry | 0 | 0 | 100 | 0 | 0 | 100 |
| 012 | | 013_apple | 10 | 10 | 80 | 20 | 0 | 80 |
| 013 | | 014_lemon | 0 | 0 | 100 | 0 | 0 | 100 |
| 014 | | 015_peach | 10 | 0 | 90 | 0 | 0 | 100 |
| 015 | | 016_pear | 10 | 0 | 90 | 0 | 10 | 90 |
| 016 | | 017_orange | 10 | 0 | 90 | 0 | 0 | 100 |
| 017 | | 018_plum | 0 | 0 | 100 | 0 | 0 | 100 |
| 019 | | 043_phillips_screwdriver | 0 | 100 | 0 | 0 | 100 | 0 |
| 020 | | 043_flat_screwdriver | 0 | 100 | 0 | 0 | 100 | 0 |
| 021 | | 057_racquetball | 0 | 0 | 100 | 0 | 0 | 100 |
| 024 | | 072_a_toy_airplane | 20 | 70 | 10 | 0 | 100 | 0 |
| 035 | | jvr_cleansing_foam | 0 | 100 | 0 | 10 | 90 | 0 |
| 036 | | dabao_wash_soup | 50 | 30 | 20 | 0 | 100 | 0 |
| 037 | | nzskincare_mouth_rinse | 10 | 70 | 20 | 0 | 100 | 0 |
| 038 | | dabao_sod | 10 | 80 | 10 | 60 | 40 | 0 |
| 039 | | soap_box | 20 | 10 | 70 | 80 | 0 | 20 |
| 040 | | kispa_cleanser | 0 | 70 | 30 | 0 | 100 | 0 |
| 041 | | darlie_toothpaste | 0 | 90 | 10 | 0 | 100 | 0 |
| 043 | | baoke_marker | 0 | 100 | 0 | 0 | 100 | 0 |
| 044 | | hosjam | 10 | 80 | 10 | 0 | 100 | 0 |
| 058 | | darlie_box | 0 | 100 | 0 | 0 | 100 | 0 |
| 059 | | soap | 10 | 40 | 50 | 50 | 20 | 30 |
| 061 | | dabao_facewash | 0 | 80 | 20 | 0 | 100 | 0 |
| 062 | | pantene | 20 | 70 | 10 | 0 | 100 | 0 |
| 063 | | head_shoulders_supreme | 20 | 80 | 0 | 0 | 100 | 0 |
| 064 | | thera_med | 10 | 80 | 10 | 0 | 100 | 0 |
| 065 | | dove | 20 | 40 | 40 | 0 | 100 | 0 |
| 066 | | head_shoulders_care | 0 | 90 | 10 | 0 | 100 | 0 |

of 16 actuators, four in each finger. The hand is controlled through ROS which also provides data stream of joint angles and torques. Here also, the locations of the finger tips are computed using the kinematics (9) with extracted joint angles. In addition, we compute the contacts force directions using angles and torques according to (10). Grasps of various objects with the Allegro hand during experiments can be seen in Figures 1 and 20. Figures 21-24 (see also Extension 1) show snapshots of grasp iterations for four YCB objects and the certainty about the object increasing when using BC-NP ($z = 3$).

Experiments with both hands are performed after training classifiers using only the CAD models of the 10 objects as described in Section 4. Here also, the classifiers are independent of the robotic hands to be used. In both the simulated and real hands, we acquire the position of the finger tips through forward kinematics. However, it is challenging to exactly measure the locations of the contact points. This tends to confuse a classifier trained over exact positions on the object. Hence, we add Gaussian noise with zero mean to the contact locations of the CAD models during the generation of training data. Consequently, the classifier becomes more robust and immune to inaccuracies in contact locations. While in our experiments it was shown unnecessary, bias can also be added to the noise in the training for better robustness.

Classification results for 10 queries per object can be seen in Table 8. In each each query, grasp samples are taken randomly by arbitrary choosing approach angles of the hand to grasp the object. When using a robotic hand, the distribution of contact points is limited based on its kinematics. Hence, the number of iterations required for sufficient certainty is higher. The success rates for BC-NP are the highest while requiring relatively many grasp samples. Nevertheless, when using $z$-finger grasps of $z = 3$, the number of iterations is reduced with the cost of a slightly lower accuracy. Here also, IC with $z = 3$ result in inaccurate classification. The use of normals at the contact points acquired from the Allegro hand did not improve accuracy much due to noisy torque measurements. Nevertheless, such addition of data enabled the reduction of the number of iterations. While five or six grasps can be seen as too much, we note that grasps do not have to be significantly different. In our experiments, sliding of a finger from a former grasp configuration to a near region was also considered as a new grasp sample. Examples for this can be seen in Figures 21-24. Thus, sampling can be fast and efficient, very similar to human tactile motion.

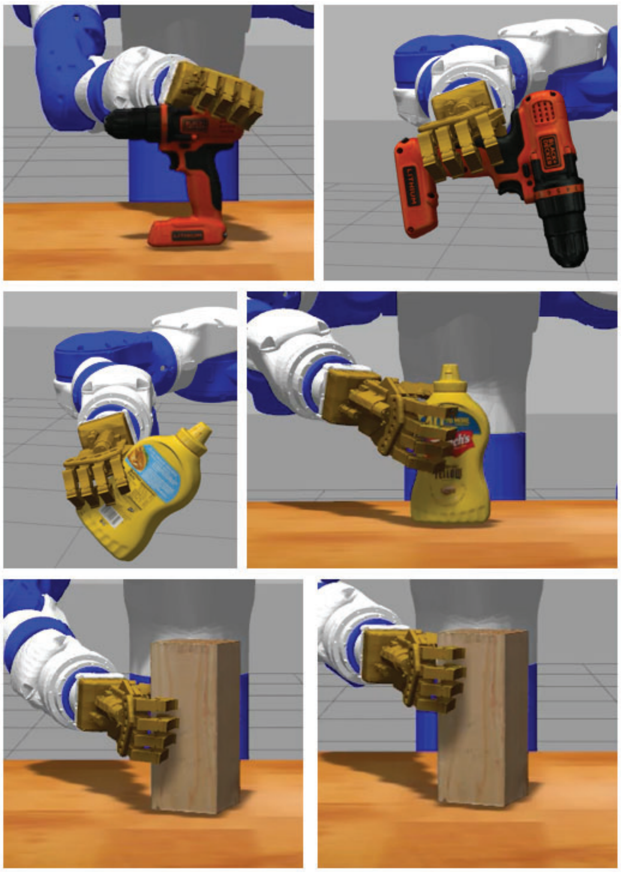


Figure 19: Three YCB objects grasped by the 5-finger Pisa/IIT soft hand in various grasps: (top) power drill, (middle) Mustard bottle and (bottom) wood block.

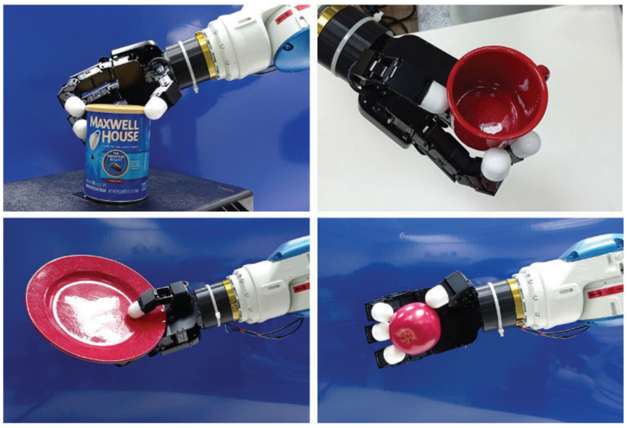


Figure 20: Object recognition experiments with the 4-finger Allegro hand. The images show grasp instances of four YCB objects.

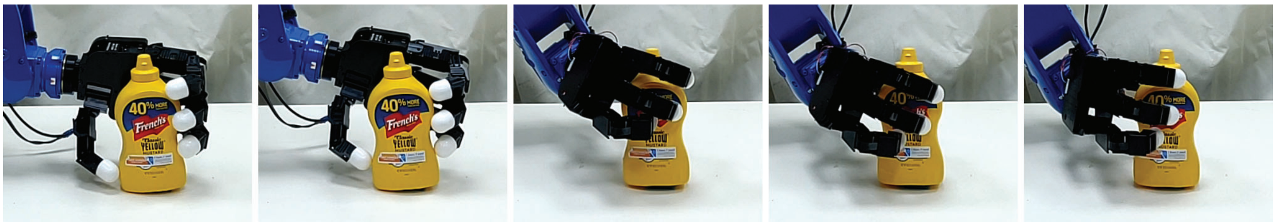


Figure 21: Five iterative grasps of the Mustard Bottle object in an experiment with the 4-finger Allegro hand. The recognition certainties about the object using BC-NP ($z = 3$) without normals are, from left to right, $P_1(\mathcal{O}_{\text{mustard}}|\mathbf{q}_1) = 0.36$, $P_2(\mathcal{O}_{\text{mustard}}|\mathbf{q}_1, \mathbf{q}_2) = 0.51$, $P_2(\mathcal{O}_{\text{mustard}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = 0.66$, $P_4(\mathcal{O}_{\text{mustard}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = 0.84$ and $P_5(\mathcal{O}_{\text{mustard}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = 0.90$.

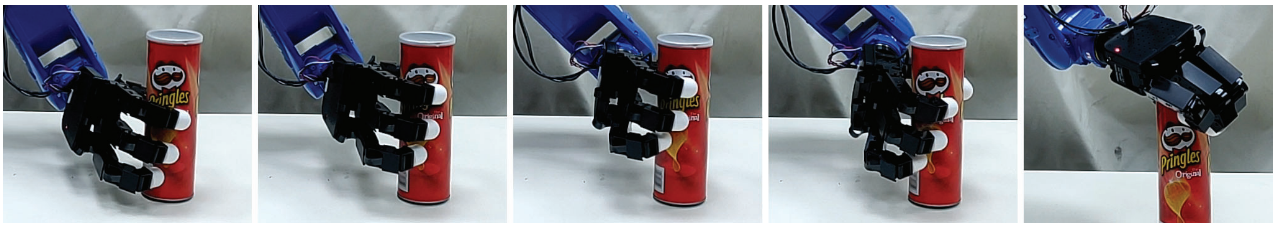


Figure 22: Five iterative grasps of the Chips Can object in an experiment with the 4-finger Allegro hand. The recognition certainties about the object using BC-NP ($z = 3$) without normals are, from left to right, $P_1(\mathcal{O}_{\text{chips}}|\mathbf{q}_1) = 0.23$, $P_2(\mathcal{O}_{\text{chips}}|\mathbf{q}_1, \mathbf{q}_2) = 0.39$, $P_2(\mathcal{O}_{\text{chips}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = 0.63$, $P_4(\mathcal{O}_{\text{chips}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = 0.80$ and $P_5(\mathcal{O}_{\text{chips}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = 0.86$.

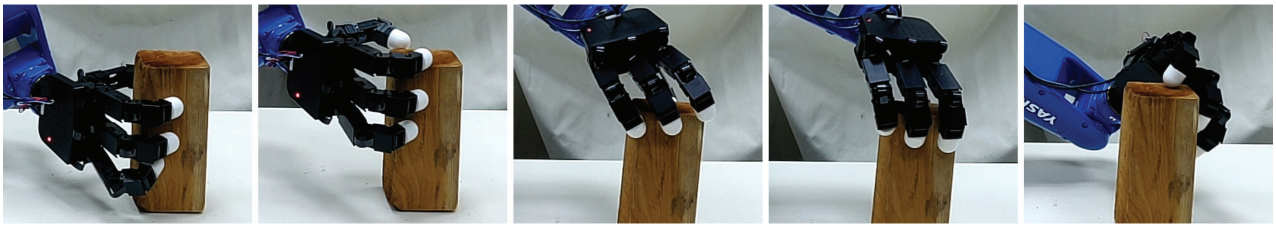


Figure 23: Five iterative grasps of the Wood Block object in an experiment with the 4-finger Allegro hand. The recognition certainties about the object using BC-NP ($z = 3$) without normals are, from left to right, $P_1(\mathcal{O}_{\text{wood}}|\mathbf{q}_1) = 0.20$, $P_2(\mathcal{O}_{\text{wood}}|\mathbf{q}_1, \mathbf{q}_2) = 0.18$, $P_2(\mathcal{O}_{\text{wood}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = 0.12$, $P_4(\mathcal{O}_{\text{wood}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = 0.81$ and $P_5(\mathcal{O}_{\text{wood}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5) = 0.89$.

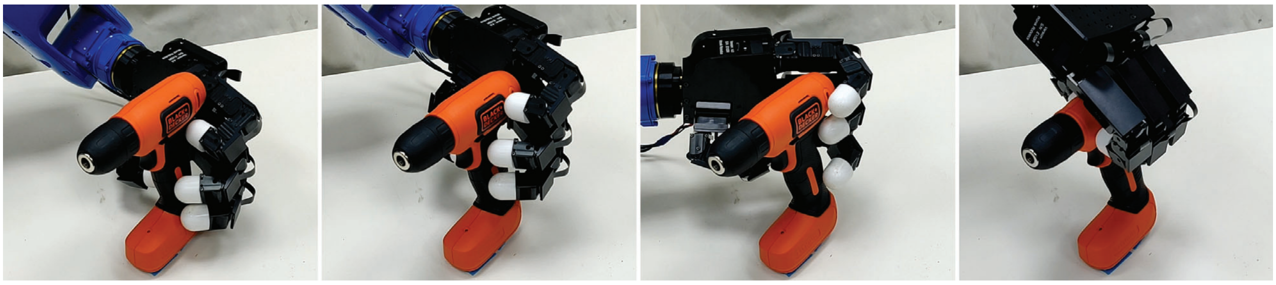


Figure 24: Four iterative grasps of the Power Drill object in an experiment with the 4-finger Allegro hand. The recognition certainties about the object using BC-NP ($z = 3$) without normals are, from left to right, $P_1(\mathcal{O}_{\text{drill}}|\mathbf{q}_1) = 0.29$, $P_2(\mathcal{O}_{\text{drill}}|\mathbf{q}_1, \mathbf{q}_2) = 0.33$, $P_2(\mathcal{O}_{\text{drill}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = 0.51$ and $P_4(\mathcal{O}_{\text{drill}}|\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4) = 0.87$.

Table 8: Results for real hands experiments

| | | Pisa hand | Allegro hand | |
|---|---|---|---|---|
| n | | 5 | 4 | |
| with/without normals | | w/o | w/o | w/ |
| IC | success rate | 89% | 81% | 87% |
| | avg. num. of iteration | 3.76 | 2.97 | 3.78 |
| BC-NP | success rate | 94% | 94% | 92% |
| | avg. num. of iteration | 8.15 | 10.45 | 4.80 |
| BC-IP | success rate | 90% | 92% | 93% |
| | avg. num. of iteration | 4.86 | 9.18 | 4.17 |
| IC, $z = 3$ | success rate | 51% | 52% | 49% |
| | avg. num. of iteration | 1.04 | 1.25 | 1.14 |
| BC-NP, $z = 3$ | success rate | 92% | 93% | 94% |
| | avg. num. of iteration | 4.22 | 4.85 | 1.62 |

# 7 Conclusions

In this paper, we have proposed a novel approach for object classification through haptic glances without the use of tactile sensors. The approach is based on a unique representation of a grasp of an object. The grasp is represented as polyhedron formed by the contact points and described by a unique frame invariant parameterization. Grasp parameterization samples are solely taken from CAD models of a set of objects and used as training data for a classifier. A grasp of an object is an instance of the object's shape and may not embed sufficient information for a certain classification. Hence, we propose, observe and compare between two iterative methods where additional grasps are used to improve classification certainty. The approach was shown to be accurate both in a thorough analysis and in real hand experiments.

The proposed method is limited to rigid objects with available CAD models. If such a model cannot be acquired for an object, one can collect real-world data through actual sampling of grasps with a robotic hand. While the collection is performed with a specific $n$-finger hand, the samples can be used to learn recognition with any $n_l$-finger hand where $2 \leq n_l \leq n$. Future work may consider data efficient collection of grasp samples by having a probabilistic approximation (e.g., Bayesian inference) of the shape. Then, a recognition classifier can be trained while considering parameterization vector deviations within the probability distributions. Similarly, deviation in the parameterization vector could be considered when dealing with soft or deformable objects and finger pose variations may be large.

Future work may also include smart and informed haptic glances where individual fingers are moved in planned directions to improve classification certainty. The iterative algorithms can also integrate haptic glances from other sources such as visual perception and tactile sensors. In underactuated hands where the kinematics are not known, tactile sensors can be used along the fingers to identify where contact occurs and acquire multiple combinations of haptic glances for predictions. Hence, one grasp can provide several sequences of grasps and may remove the need to sample another grasp.

# References

Allen PK and Roberts KS (1989) Haptic object recognition using a multi-fingered dextrous hand. In: *Proceedings, 1989 International Conference on Robotics and Automation*. pp. 342–347 vol.1.

Avis D and El-Gindy H (1987) Triangulating point sets in space. *Discrete & Computational Geometry* 2: 99–111.

Azulay O, Ben-David I and Sintov A (2023) Learning haptic-based object pose estimation for in-hand manipulation control with underactuated robotic hands. *IEEE Trans. Haptics* 16(1): 73–85.

Besl P and McKay ND (1992) A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2): 239–256.

Bhattacharjee T, Rehg JM and Kemp CC (2012) Haptic classification and recognition of objects using a tactile sensing forearm. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 4090–4097.

Björkman M, Bekiroglu Y, Högman V and Kragic D (2013) Enhancing visual perception of shape through tactile glances. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3180–3186.

Bradford-Barber C, Dobkin DP and Huhdanpaa H (1996) The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22(4): 469–483.

Calli B, Singh A, Bruce J, Walsman A, Konolige K, Srinivasa S, Abbeel P and Dollar AM (2017) Yale-CMU-Berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research* 36(3): 261–268.

Carter J and Fourney D (2005) Research based tactile and haptic interaction guidelines. In: *Proceedings of Guidelines On Tactile and Haptic Interactions*. pp. 84–92.

Catalano M, Grioli G, Farnioli E, Serio A, Piazza C and Bicchi A (2014) Adaptive synergies for the design and control of the Pisa/IIT softhand. *The International Journal of Robotics Research* 33(5): 768–782.

Charles RQ, Su H, Kaichun M and Guibas LJ (2017) PointNet: Deep learning on point sets for 3D classification and segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 77–85.

Dario P, Ferrante P, Giacalone G, Livaldi L, Allotta B, Buttazzo G and Sabatini AM (1992) Planning and executing tactile exploratory procedures. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3. pp. 1896–1903.

Deneen L and Shute G (1988) Polygonizations of point sets in the plane. *Discrete & Computational Geometry* 3: 77–87.

Deng J, Dong W, Socher R, Li LJ, Li K and Li FF (2009) Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition* : 248–255.

Dragiev S, Toussaint M and Gienger M (2013) Uncertainty aware grasping and tactile exploration. In: *IEEE International Conference on Robotics and Automation*. pp. 113–119.

Driess D, Englert P and Toussaint M (2017) Active learning with query paths for tactile object shape exploration. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 65–72.

Drimus A, Kootstra G, Bilberg A and Kragic D (2011) Classification of rigid and deformable objects using a novel tactile sensor. In: *International Conference on Advanced Robotics*. pp. 427–434.

Drimus A, Kootstra G, Bilberg A and Kragic D (2014) Design of a flexible tactile sensor for classification of rigid and deformable objects. *Robotics and Autonomous Systems* 62(1): 3 – 15.

Du G, Wang K, Lian S and Zhao K (2020) Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review* .

Fang HS, Wang C, Gou M and Lu C (2020) Graspnet-1billion: A large-scale benchmark for general object grasping. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 11444–11453.

Fishel J and Loeb G (2012) Bayesian exploration for intelligent identification of textures. *Frontiers in neurorobotics* 6: 4.

Griffiths D and Boehm J (2019) A review on deep learning techniques for 3D sensed data classification. *Remote. Sens.* 11: 1499.

Han X, Li Z, Huang H, Kalogerakis E and Yu Y (2017) High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Hendrycks D and Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: *International Conference on Learning Representations (ICLR)*, volume abs/1610.02136.

Iwamura Y, Tanaka M, Hikosaka O and Sakamoto M (1995) Postcentral neurons of alert monkeys activated by the contact of the hand with objects other than the monkey's own body. *Neuroscience Letters* 186(2): 127 – 130.

Jamali N, Ciliberto C, Rosasco L and Natale L (2016) Active perception: Building objects' models using tactile exploration. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. pp. 179–185.

James JW, Church A, Cramphorn L and Lepora NF (2020) Tactile model O: Fabrication and testing of a 3D-printed, three-fingered tactile robot hand. *Soft Robotics* .

Jin M, Gu H, Fan S, Zhang Y and Liu H (2013) Object shape recognition approach for sparse point clouds from tactile exploration. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. pp. 558–562.

Kahanowich ND and Sintov A (2021) Robust classification of grasped objects in intuitive human-robot collaboration using a wearable force-myography device. *IEEE Robotics and Automation Letters* 6(2): 1192–1199.

Kasper A, Xue Z and Dillmann R (2012) The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research* 31(8): 927–934.

Kay SM (1993) *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc.

Klatzky R and Lederman S (1995) Identifying objects from a haptic glance. *Attention Perception & Psychophysics* 57: 1111–1123.

Klatzky RL, Lederman SJ and Metzger VA (1985) Identifying objects by touch: An "expert system". *Perception & psychophysics* 37(4): 299–302.

Lederman SJ and Klatzky RL (1987) Hand movements: A window into haptic object recognition. *Cognitive Psychology* 19(3): 342 – 368.

Lederman SJ and Klatzky RL (1993) Extracting object properties through haptic exploration. *Acta Psychologica* 84(1): 29 – 40. Tactile Pattern Recognition.

Li L, Hu Q, Wu X and Yu D (2014) Exploration of classification confidence in ensemble learning. *Pattern Recognition* 47(9): 3120–3131.

Li Y and Pollard NS (2005) A shape matching algorithm for synthesizing humanlike enveloping grasps. In: *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*. pp. 442–449.

Lin J, Calandra R and Levine S (2019) Learning to identify object instances by touch: Tactile recognition via multimodal matching. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 3644–3650.

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick CL (2014) Microsoft COCO: Common objects in context. In: Fleet D, Pajdla T, Schiele B and Tuytelaars T (eds.) *Computer Vision – ECCV*. pp. 740–755.

Liu H, Song X, Nanayakkara T, Seneviratne LD and Althoefer K (2012) A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor. In: *IEEE International Conference on Robotics and Automation*. pp. 1410–1415.

Liu H, Wu Y, Sun F and Guo D (2017) Recent progress on tactile object recognition. *International Journal of Advanced Robotic Systems* 14(4).

Lowe DG (1999) Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2. pp. 1150–1157 vol.2.

Luo S, Mou W, Althoefer K and Liu H (2018) iCLAP: shape recognition by combining proprioception and touch sensing. *Autonomous Robots* .

Morgan AS, Hang K, Wen B, Bekris KE and Dollar AM (2022) Complex in-hand manipulation via compliance-enabled finger gaiting and multi-modal planning. *IEEE Robotics Autom. Lett.* 7(2): 4821–4828.

Murray RM, Li Z and Sastry SS (1994) *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Museros L, Falomir Z, Velasco F, Gonzalez-Abril L and Martí I (2012) 2D qualitative shape matching applied to ceramic mosaic assembly. *Journal of Intelligent Manufacturing* 23(5): 1973–1983.

Narang YS, Wyk KV, Mousavian A and Fox D (2020) Interpreting and predicting tactile signals via a physics-based and data-driven framework. *CoRR* .

Nelinger G, Assa E and Ahissar E (2016) *Tactile Object Perception*. Paris: Atlantis Press, pp. 161–176.

Novotni M and Klein R (2001) A geometric approach to 3D object comparison. *International Conference on Shape Modeling and Applications* : 154–166.

Ohbuchi R, Otagiri T, Ibato M and Takei T (2002) Shape-similarity search of three-dimensional models using parameterized statistics. In: *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*. pp. 265–274.

Okamura AM, Turner ML and Cutkosky MR (1997) Haptic exploration of objects with rolling and sliding. In: *IEEE International Conference on Robotics and Automation*, volume 3. pp. 2485–2490 vol.3.

OpenAI, Akkaya I, Andrychowicz M, Chociej M, Litwin M, McGrew B, Petron A, Paino A, Plappert M, Powell G, Ribas R, Schneider J, Tezak N, Tworek J, Welinder P, Weng L, Yuan Q, Zaremba W and Zhang L (2019) Solving rubik's cube with a robot hand. *CoRR* abs/1910.07113. URL http://arxiv.org/abs/1910.07113.

Osada R, Funkhouser T, Chazelle B and Dobkin D (2001) Matching 3D models with shape distributions. In: *Proceedings International Conference on Shape Modeling and Applications*. pp. 154–166.

Osada R, Funkhouser T, Chazelle B and Dobkin D (2002) Shape distributions. *ACM Transactions on Graphics* 21(4): 807–832.

Parzen E (1962) On estimation of a probability density function and mode. *Ann. Math. Statist.* 33(3): 1065–1076.

Preparata FP and Shamos MI (1985) *Computational Geometry*. New York: Springer-Verlag.

Rouhafzay G and Cretu A (2020) Object recognition from haptic glance at visually salient locations. *IEEE Transactions on Instrumentation and Measurement* 69(3): 672–682.

Sarkar K, Varanasi K and Stricker D (2017) Learning quadrangulated patches for 3D shape parameterization and completion. In: *International Conference on 3D Vision (3DV)*. pp. 383–392.

Sinapov J, Bergquist T, Schenck C, Ohiri U, Griffith S and Stoytchev A (2011) Interactive object recognition using proprioceptive and auditory feedback. *The International Journal of Robotics Research* 30(10): 1250–1262.

Sintov A, Menassa R and Shapiro A (2016) A gripper design algorithm for grasping a set of parts in manufacturing lines. *Mechanism and Machine Theory* 105: 1 – 30.

Sintov A, Menassa RJ and Shapiro A (2014) OCOG: A common grasp computation algorithm for a set of planar objects. *Robotics and Computer-Integrated Manufacturing* 30(2): 124 – 141.

Sintov A, Raghothama S, Menassa R and Shapiro A (2012) A common 3-finger grasp search algorithm for a set of planar objects. In: *Proceeding of the IEEE International Conference on Automation Science and Engineering (CASE)*. Seoul, Korea, pp. 1091–1096.

Sintov A and Shapiro A (2015) Automatic design algorithm of a robotic end-effector for a set of sheet-metal parts. In: *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. Angkor-Wat, Cambodia, pp. 19–24.

Spiers AJ, Liarokapis MV, Calli B and Dollar AM (2016) Single-grasp object classification and feature extraction with simple robot hands and tactile sensors. *IEEE Transactions on Haptics* 9(2): 207–220.

Su H, Maji S, Kalogerakis E and Learned-Miller E (2015) Multi-view convolutional neural networks for 3d shape recognition. In: *IEEE International Conference on Computer Vision (ICCV)*. pp. 945–953.

Su Z, Fishel J, Yamamoto T and Loeb G (2012) Use of tactile feedback to control exploratory movements to characterize object compliance. *Frontiers in neurorobotics* 6: 7.

Thrun S, Burgard W and Fox D (2005) *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press. ISBN 0262201623.

Torres-Jara E, Natale L and Fitzpatrick P (2005) Tapping into touch. In: *International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*.

Yan X, Lin L, Mitra NJ, Lischinski D, Cohen-Or D and Huang H (2022) Shapeformer: Transformer-based shape completion via sparse representation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 6239–6249.

Yi Z, Calandra R, Veiga F, van Hoof H, Hermans T, Zhang Y and Peters J (2016) Active tactile object exploration with gaussian processes. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4925–4930.

Yuan W, Wang S, Dong S and Adelson E (2017) Connecting look and feel: Associating the visual and tactile properties of physical materials. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 4494–4502.

Zhou L, Du Y and Wu J (2021) 3d shape generation and completion through point-voxel diffusion. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 5806–5815.

# Appendix A: Index to Multimedia Extensions

Table 9: *
Table of Multimedia extensions

| Extension | Media type | Description |
|---|---|---|
| 1 | Video | Experiments where grasp iterations with the 4-finger Allegro hand are used to recognize several YCB objects. |

# Appendix B: Proof of Theorems

Proof of Theorem 1 is as follows.

*Proof.* Rotation $R$ and translation $\mathbf{d}$ represent a rigid transformation in the Euclidean space and therefore, preserves distance between points and angles between vectors (Murray et al., 1994). Hence, all lengths and angles of $\Theta(\mathcal{P}_k)$ are equal to $\Theta(R\mathcal{P}_k + \mathbf{d})$. Similarly, the angles between the vectors in $\mathcal{N}_k$ to edges of $\Theta(\mathcal{P}_k)$ are equal to the angles between the vectors in $R\mathcal{N}_k$ to edges of $\Theta(R\mathcal{P}_k + \mathbf{d})$. Consequently, both grasps are defined by the same polyhedron and will be parameterized equally by map $\Phi_n$. □

Next, we present the proof for Theorem 3.

*Proof.* Scaling of the polyhedron $\Theta(\mathcal{P}_i)$ by $\xi$ yields a new polyhedron $\Theta(\mathcal{P}_j)$, where $\mathcal{P}_i \in \mathcal{G}_i$ and $\mathcal{P}_j = \xi \cdot \mathcal{P}_i \in \mathcal{G}_j$, such that $\Theta(\mathcal{P}_i)$ and $\Theta(\mathcal{P}_j)$ are similar (Osada et al., 2001). Hence, all matching angles between both polyhedrons are equal. Furthermore, this imposes proportionality between any edge $e_{ik} \in \Theta(\mathcal{P}_i)$ and its corresponding edge $e_{jk} \in \Theta(\mathcal{P}_j)$, i.e., $\frac{e_{jk}}{e_{ik}} = \xi$. Also, if two polyhedrons are similar with a scale factor of $\xi$, then their surface areas are in the ratio of $\xi^2$. Therefore, for any given edge $k$, it follows that

$$\xi^2 = \left(\frac{e_{jk}}{e_{ik}}\right)^2 = \frac{\sum_{u=1}^{U} a_{ju}}{\sum_{u=1}^{U} a_{iu}} \tag{25}$$

where $a_{vu}$ is the surface area of facet $u$ ($u = 1, ..., U$) on polyhedron $\Theta(\mathcal{P}_v)$. Again, let $A_v = \sqrt{\sum_{u=1}^{U} a_{vu}}$, then it must be that

$$\frac{e_{jk}}{e_{ik}} = \frac{A_j}{A_i} \quad \text{or} \quad \frac{e_{jk}}{A_j} = \frac{e_{ik}}{A_i}, \tag{26}$$

for any two corresponding edges $e_{jk}$ and $e_{ik}$. Hence, scaling according to $\frac{1}{A_j}\mathcal{P}_j$ and $\frac{1}{A_i}\mathcal{P}_i$ will yield two equal polyhedrons. Consequently, the parameterization vectors $\Phi_n\left(\mathcal{G}_i^{\left(A_i^{-1}\right)}\right)$ and $\Phi_n\left(\mathcal{G}_j^{\left(A_j^{-1}\right)}\right)$ are equal. □