Fingertip Surface Optimization for Robust Grasping on Contact Primitives

Haoran Song, Michael Yu Wang, and Kaiyu Hang

***Abstract*—We address the problem of fingertip design by leveraging on the fact that most grasp contacts share a few classes of local geometries. In order to maximize the contact areas for achieving more robust grasps, we first define the concept of *Contact Primitive*, which represents a set of contacts of similar local geometries. Thereafter, we propose a uniform cost algorithm, which is formulated as a decision making process in a tree structure, to cluster a set of example grasp contacts into a finite set of *Contact Primitives*. We design fingertips by optimization to match the local geometry of each contact primitive, and then three-dimensionally print them using soft materials to compensate for optimization residuals. For novel objects, we provide an efficient algorithm to generate grasp contacts that match the fingertip geometries while together forming stable grasps. Comparing to a baseline of flat fingertip design, the experiment results show that our design significantly improves grasp stability, and it is more robust against various uncertainties.**

***Index Terms*—Grasping, dexterous manipulation, multifingered hands, contact modeling, additive manufacturing.**

I. Introduction

GRASPING is an essential ability that enables the robot to physically interact with the world. Contact-based grasping [1], which explicitly considers the contacts made by a grasp, has been widely studied to address various aspects of grasping [2]–[8]. Contact modeling is one of the basic building blocks for the research of contact-based grasping. Dependent on the friction and softness at contacts, form or force closure based grasp quality can be calculated in the grasp wrench space [9], [10]. However, it is commonly identified that the analytic contact modeling has two major issues: 1) the point contact model does not precisely reflect the real physics [10]; and 2) it is vulnerable to uncertainties in positioning [11], friction coefficient [12], visual perception [13], etc.

Manuscript received September 10, 2017; accepted December 24, 2017. Date of publication January 4, 2018; date of current version January 16, 2018. This letter was recommended for publication by Associate Editor A. Morales and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported in part by the HKUST SSTSP project RoMRO (FP802), in part by the HKUST IGN project IGN16EG09, and in part by the HKUST PGS Fund of Office of Vice-President (Research and Graduate Studies). *(Corresponding author: Kaiyu Hang.)*

H. Song is with the Department of Mechanical and Aerospace Engineering and the HKUST Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong (e-mail: hsongad@connect.ust.hk).

M. Y. Wang is with the Department of Mechanical and Aerospace Engineering and the Department of Electronic and Computer Engineering, and the HKUST Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong (e-mail: mywang@ust.hk).

K. Hang is with the Department of Computer Science and Engineering, HKUST Institute for Advanced Study, and the HKUST Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong (e-mail: kaiyuh@kth.se).

Digital Object Identifier 10.1109/LRA.2018.2789842

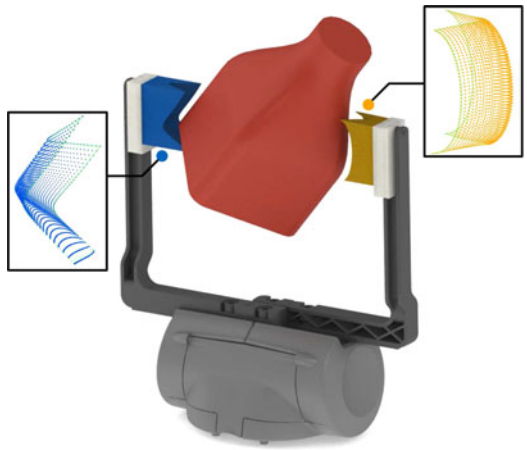

Fig. 1. An example grasp using 2 of the designed fingertips. The point clouds depict the geometries of the optimized fingertip surfaces.

To counteract the uncertainties, one of the recent research directions has focused on fingertip design to replace the commonly used flat surface. By designing the fingertips to perfectly match the local geometries of contacts, the contact areas are maximized to improve the grasp stability [14], [15]. This line of work has indicated that the modeling of contact geometries for fingertip design can significantly improve grasp stability, as well as improving the robustness against uncertainties. However, it is worthwhile noting that the existing work is incapable of addressing novel or a large number of objects.

Although objects possess a large variety of global shapes, we identify the fact that most grasp contacts on daily objects share only a few classes of local geometries, such as edges, corners, curved surfaces, etc. As such, we aim to design robotic fingertips to maximally mimic each of those local geometry classes, so as to maximize the grasp contact areas on arbitrary objects. For this, we define the concept of *Contact Primitive* to represent a set of similar contact local geometries, and optimize the corresponding fingertip surface to accommodate each geometry within the *Contact Primitive*. Given a set of training objects with desired grasp contacts, we provide a uniform cost algorithm, which is formulated as a decision making process in a tree structure, to first *optimally* cluster the contacts into a finite set of *Contact Primitives*, and then design a fingertip for each *Contact Primitive*. The designed fingertips are 3D printed using soft materials, in order to compensate for the optimization residuals to further maximize contact areas.

Moreover, we provide a hierarchical algorithm for grasp planning on novel objects to find contacts that match the designed fingertip geometries, while providing stable and reachable grasps. Fig. 1 shows an example of 2 fingertip designs, as well as the

2377-3766 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

generated grasp contacts at the areas corresponding to the *Contact Primitives*. For evaluation, we install the designed fingertips on a Baxter robot and show that they are able to grasp a large set of novel objects. We provide both quantitative results showing that the optimized fingertips significantly improve grasp stability, and qualitative examples indicating that it is more robust against uncertainties.

We review related work in Section II and define the concept of *Contact Primitive* in Section III. In Section IV we explain the fingertip optimization, followed by the description of hierarchical grasp synthesis in Section V. Finally, we describe the experiment results in Section VI and conclude in Section VII.

II. Related Work

Contact-level grasping has been studied from various aspects and shown great potential in applications that cannot be achieved using other grasping types [1], [2], [10]. On one hand, the explicit contact modeling allows for analytic force calculation [4], stability estimation [5], hand configuration computation [6], etc. On the other hand, as it uses only the fingertips for grasping, it is able to pick up small or fragile objects, and more importantly, it allows for dexterous in-hand manipulation [7], [8].

A. Contact Modeling

The flexibility of contact-level grasping, e.g., in force and kinematics control, has also raised many challenges for applying such grasps. One of the major problems is caused by the unrealistic modeling of point contact used in grasp evaluation [10], which can falsely indicate a stable grasp configuration that may fail in the real execution. In order to compensate for this imprecise contact modeling, tactile feedback has been used to predict grasp stability online after a grasp is executed [16]. By modeling the grasping dynamics in the object-level frame, tactile feedback can also be adopted for online force regulation and finger gaiting [7]. In order to avoid the potential failure caused by positioning errors, independent contact regions is proposed to introduce positioning tolerance [11]. Moreover, cage grasping has been investigated to immobilize objects without requiring precise contact locations [17].

B. Fingertip Design

From the perspective of hardware design, dexterous underactuated hands have been designed to implicitly reach for contacts and wrap around the object for providing better stability [18]. In the other research line, finger design is carried out to ensure that the fingertips can perfectly match the contact local geometries of a few objects, so as to improve grasp stability [14]. However, the existing methods are dedicated to a few pre-defined workpieces and unable to generalize to more objects. In this work, we further identify the fact that most grasp contacts share a few classes of local geometries, which can be contacted using a few fingertip designs to maximize the contact areas. Therefore, based on a set of training objects, we define the concept of *Contact Primitives* to represent different classes of contacts. As such, our fingertip design is able to generalize to much more objects, especially when they are manufactured using soft materials.

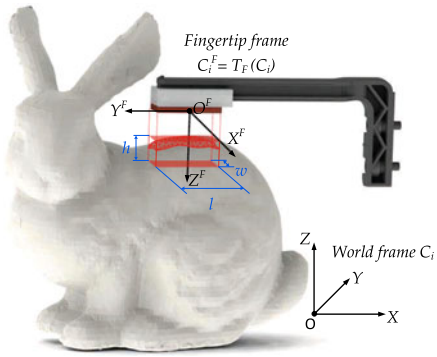

Fig. 2. Example of contact area extraction and transformation to the fingertip frame.

C. Grasp Planning

With respect to different task requirements, previous works have developed many approaches to generate grasp contacts and even with associated hand configurations [19]–[23]. However, the local geometries of contacts are rarely considered. As we aim at providing maximized contacts between fingertips and objects, when encountering novel objects, we provide an efficient hierarchical algorithm to generate grasp contacts that match the designed fingertip geometries, as well as ensuring stable and reachable grasps.

III. Contact Primitives

In this section, we introduce the concept of *Contact Primitive*. Based on an object point cloud with an example grasp, we first describe the extraction of local contacts and show how to measure the difference between contacts. Given a set of extracted contacts from different objects and grasps, we provide an algorithm to optimally cluster the set into expected number of classes and ensure the minimum sum of in-class differences. The obtained contact classes essentially yield the definition of *Contact Primitives*.

A. Contact Area Extraction

As the goal of fingertip design is to maximize the contact areas with respect to fingertip size, local contacts are extracted to maximally contain the local geometries in relation to the fingertip scale. Concretely, denoted by $\mathcal{P} \subset \mathbb{R}^3$ the point cloud of an object and $g = \{(c_1, t_1), \ldots, (c_m, t_m) \mid c_i \in \mathcal{P}, t_i \in \mathbb{S}^2\}$ the m contacts of a grasp g on $\mathcal{P}$, with t_i being the finger pointing direction. A contact area is defined as $\mathcal{C}_i \subset \mathcal{P}$ centered at c_i. Formally, the contact area is extracted as:

$$\mathcal{C}_i = \{p_i \mid p_i \in \psi(c_i, t_i) \ \wedge \ p_i \in \mathcal{P}\} \tag{1}$$

where $\psi(c_i, t_i)$ denotes a cuboid area centered at c_i with width w and length l, which are the width and length of the fingertip. The cuboid area also has a depth h and is oriented to have the Y^F direction aligned with the finger pointing direction t_i and Z^F direction the same as the fingertip's normal. As shown in Fig. 2, the contact area is extracted from $\mathcal{P}$ as a projection of the fingertip shape. Note that the above definition can be adapted to arbitrary fingertips and is not limited to the cuboid shape, as long as the fingertip shape can be projected to the object surface as described.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

We can see from the above definition that the contact area is expressed in the world frame. However, in order to allow the comparison between contact areas, the extracted areas should be oriented so that they always align with the fingertip. In other words, if we want to compare two contact areas, they should be aligned in the same orientation as the fingertip contacts them. For this, we transform the extracted $\mathcal{C}_i$ into the fingertip frame as below:

$$\mathcal{C}_i^F = T_F(\mathcal{C}_i), \quad \mathcal{C}_i^F \subset \mathbb{R}^3 \tag{2}$$

where $T_F(\cdot)$ is the transformation from the world frame to the fingertip frame. Fig. 2 shows an example of how a contact area is transformed to the fingertip frame.

Having obtained the contact areas from example grasps, we also wish to measure the difference between them. For this, given any two contact areas $\mathcal{C}_i^F$ and $\mathcal{C}_j^F$ expressed in the fingertip frame, assuming that $|\mathcal{C}_i^F| \leq |\mathcal{C}_j^F|$, we define the difference measure as:

$$\gamma(\mathcal{C}_i^F, \mathcal{C}_j^F) = \frac{\sum_{p_i \in \mathcal{C}_i^F} (p_i^z - p_j^z)^2}{|\mathcal{C}_i^F|} \tag{3}$$

where $p_j \in \mathcal{C}_j^F$ is the nearest point to p_i in $\mathcal{L}^2$ distance, $|\mathcal{C}_i^F|$ and $|\mathcal{C}_j^F|$ are the cardinalities of the point sets. p_i^z and p_j^z denote the z component of the coordinates.

B. Contact Clustering

In order to guide the fingertip design, we have to acquire geometric properties of contact areas. Recall that most contact areas share just a few classes of local geometries, we now need to decide how to obtain and represent the local geometry classes. Firstly, we need a set of training objects associated with example grasps to produce a set of example contact areas $\mathbb{C} = \{\mathcal{C}_1^F, \ldots, \mathcal{C}_N^F\}$. Based on $\mathbb{C}$, we define the concept of *Contact Primitive*:

Definition III.1: A Contact primitive Q_i is a set of local contact areas expressed in the fingertip frame:

$$\begin{aligned} & Q_i \subset \mathbb{C} \\ \text{s.t.} \quad & Q_i \cap Q_j = \emptyset \wedge \\ & Q_1 \cup Q_2 \cup \ldots \cup Q_K = \mathbb{C} \end{aligned}$$

where K is the total number of contact primitives.

As one can observe, a contact primitive is a subset of $\mathbb{C}$ by definition. However, it should be noted that the construction of contact primitives directly affects the quality of geometric properties represented by them. A poor construction may result in a set of contact areas that are very different from each other, and is not able to jointly represent any geometric properties. As such, we are now interested in the problem of clustering $\mathbb{C}$ into K classes to compose K contact primitives, and the goal of this clustering is to ensure that the sum of the K in-class differences is minimized. Formally, the in-class difference is calculated by:

$$\Gamma(Q_i) = \sum_{\substack{\mathcal{C}_j, \mathcal{C}_k \in Q_i \\ j \neq k}} \gamma(\mathcal{C}_j^F, \mathcal{C}_k^F) \tag{4}$$

In order to optimally cluster $\mathbb{C}$, we formulate this problem as a decision making process similar to [24]. The clustering process is represented as a tree as exemplified in Fig. 3. Concretely, the root of the tree is the state in which all contact areas are not assigned to any class. Starting from the root, the edges connecting the root and the nodes in layer 1 are the decisions which assign the first contact area $\mathcal{C}_1^F \in \mathbb{C}$ to one of the classes. Once a decision is made, the edges connecting the selected node and layer 2 concern our decision on the class assignment of $\mathcal{C}_2^F \in \mathbb{C}$.

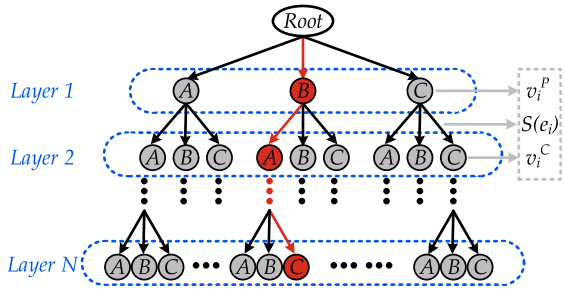

Fig. 3. An example tree for clustering N contact areas into 3 contact primitives: A, B and C. The red path shows a clustering path.

Algorithm 1: Contact Areas Clustering.

Input: $K, G = (E, V)$
Output: π^*
1: $(v_i, \Gamma(v_i)) \leftarrow (v_0, 0)$, $\mathcal{Q} \leftarrow \emptyset$ ▷ Initialization
2: **while** v_i is not a leaf **do** ▷ Main Loop
3: **for all** $e_j \in v_i$. SuccessorEdges() **do**
4: $\Gamma(v_j^C) \leftarrow \Gamma(v_i) + \mathcal{S}(e_j)$
5: $\mathcal{Q}$.push$(v_j^C, \Gamma(v_j^C))$
6: **end for**
7: $(v_i, \Gamma(v_i)) \leftarrow \mathcal{Q}$.pop()
8: **end while**
9: $\pi^* = v_i.BackTrace()$ ▷ Return Path

As this process progresses, we will finally reach the bottom layer N where each contact area has been assigned to one of the classes. Formally, the tree structure can be written as $G = (E, V)$, where V is the set of nodes denoting the current clustering state, while E denotes the edges along which decisions are made. For an edge $e_i \in E$, we define a cost as:

$$\mathcal{S}(e_i) = \Gamma(v_i^C) - \Gamma(v_i^P) \tag{5}$$

where v_i^C and v_i^P are the child and parent nodes connected by e_i. To this end, the contact area clustering problem is reduced to an optimal path finding problem in G with the root being the start position, while the goal position is one of the leaf nodes. The objective is to find a path $\pi^* = (\pi_1, \ldots, \pi_N)$ in G, $\pi_i \in E$, such that the sum of the path cost $\sum_{\pi_i \in \pi^*} \mathcal{S}(\pi_i)$ is minimized.

Once the optimal solution is found for this path finding problem, the class assigned to each contact area can be obtained by back-tracing through π^*. In this work, we employ the Dijkstra's algorithm, which is a uniform cost search algorithm to solve the reduced optimal path finding problem. This algorithm has the worst-case time complexity of $\mathcal{O}(|E| + |V| \log |V|)$. Denoted by $\mathcal{Q}$ a priority queue, the clustering process is summarized in Algorithm. 1.

To determine the number of classes K, which is the number of contact primitives, we employ the Elbow method in terms of the average in-class difference over all contact primitives: $W = \frac{\sum_{i=1}^{K} \Gamma(Q_i)}{\sum_{i=1}^{K} |Q_i|(|Q_i|-1)}$. K is selected at the turning point of the

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

W curve [25]. In our experiments, K can be determined as 3 or 4 dependent on the provided training objects and grasps.

IV. Fingertip Surface Optimization

Once the example contact areas $\mathbb{C}$ are clustered into contact primitives $\{Q_1, \ldots, Q_K\}$, as the sum of in-class differences are minimized, the contact areas contained in each contact primitive are jointly representing a specific type of local contact geometry. Therefore, it is now possible to carry out the fingertip surface optimization for each contact primitive, such that the contact areas between the fingertip and all member contact areas are maximized.

A. Surface Parameterization

For fingertip optimization, we adopt the Bézier surface [26], which is a parametric representation of continuous surfaces, to model the fingertip surface. Briefly, a d-dimensional Bézier surface of degree (m, n) is determined by the interpolation of a set of $(m+1)(n+1)$ d-dimensional control points $\rho_{ij} \in \rho \subset \mathbb{R}^d$, $i \in \{0, 1, \ldots, m\}, j \in \{0, 1, \ldots n\}$. For any point $\omega(u, v) \in \mathbb{R}^d$ on the surface, the position is determined by a function of the parametric coordinates u and v varying between 0 and 1:

$$\omega(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u) B_j^n(v) \rho_{ij} \tag{6}$$

where $B_i^m(u)$ is a Bernstein polynomial:

$$B_i^m(u) = \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i} \tag{7}$$

When the parametric coordinate u or v equals to either 0 or 1, the control points are exactly located at the edges of the bounded surface. As described in Section III-A, in terms of the fingertip shape, the contact areas are extracted within a bounded area of width w and length l, the Bézier surface is therefore bounded by:

$$\begin{aligned} \rho_{00}^x = \rho_{m0}^x = -\frac{w}{2}, \qquad & \rho_{0n}^x = \rho_{mn}^x = \frac{w}{2} \\ \rho_{00}^y = \rho_{0n}^y = -\frac{l}{2}, \qquad & \rho_{m0}^y = \rho_{mn}^y = \frac{l}{2} \end{aligned} \tag{8}$$

where ρ_{ij}^x and ρ_{ij}^y define the x-coordinate and y-coordinate for boundary control points. Note that when the fingertip is not of cuboid shapes, the boundary condition should be adpated to constrain the surface shape. Next, we need to solve a constrained programming problem to obtain a Bézier surface that maximally mimic a contact area $\mathcal{C}_i^F$:

$$\rho^* = \arg\min_{\rho} \gamma(\tau(\rho), \mathcal{C}_i^F) \text{ s.t. (8)} \tag{9}$$

where $\tau(\rho)$ denotes the surface parametrized by ρ and is discretized for the calculation of $\gamma(\cdot)$.

B. Optimization for Contact Primitives

In order to design a fingertip that maximally mimic the geometries in a contact primitive, the surface optimization has to make a tradeoff among all contained contact areas. Concretely, we adopt a 3D Bézier surface of degree $(3, 3)$, which uses $4 \times 4 = 16$ control points to parametrize the surface on a fingertip. Similar to surface optimization for a contact area defined in (9), the surface optimization problem for a contact primitive Q_j is given by:

$$\rho^* = \arg\min_{\rho} \sum_{\mathcal{C}_i^F \in Q_j} \gamma(\tau(\rho), \mathcal{C}_i^F) \text{ s.t. (8)} \tag{10}$$

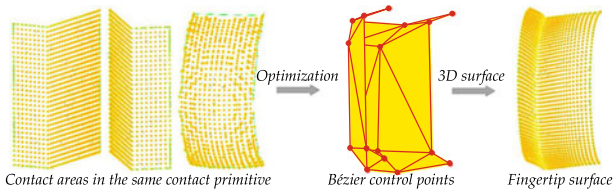

Fig. 4. An example showing the fingertip design for a contact primitive with 3 contact areas.

We solve this constrained programming problem using Sequential Quadratic Programming [27], which although does not guarantee a global optimal solution, but turns out to be very efficient to produce satisfactory solutions.

As exemplified in Fig. 4, given a contact primitive consisting of 3 contact areas, the optimized Bézier surface can well mimic the geometries to maximize the contact areas. Nevertheless, it can be observed that there are optimization residuals with respect to each individual contact area. However, as will be seen in Section VI, we 3D print the optimized fingertips using soft materials, so that the optimization residuals can be well compensated to make maximal contacts.

V. Grasp Planning

Based on a set of training objects with example grasps, we have constructed contact primitives $\{Q_1, \ldots Q_K\}$ and optimized the fingertip surface for each Q_i. It is straightforward that the example grasps can be well executed using the designed fingertips. However, since most grasp contacts share these identified classes of contact geometries, we wish to generalize the designed fingertips for grasping novel objects. To this end, in terms of the fingertip designs, we provide a grasp planning algorithm to find grasp contacts that maximally match the fingertip geometries, while forming a stable and kinematically feasible grasp.

For planning contact-based grasping, force closure is commonly adopted to calculated grasp quality. However, in order to make grasp contacts using the designed fingertips, which are not flat surfaces as on normal hands, the grasp planning is also constrained by the geometry matching. As such, there are 3 considerations involved in the grasp planning: 1) The grasp has to be stable; 2) The grasp has to be kinematically feasible for the adopted hand; and finally 3) The planned contacts have to match the geometries of designed fingertips.

Same as in Section III, let $g = \{(c_1, t_1), \ldots, (c_m, t_m) \mid c_i \in \mathcal{P}, t_i \in \mathbb{S}^2\}$ be the m contacts of a grasp, where c_i is the contact location and t_i is the finger pointing direction. The grasp planning problem can be addressed using the *Hierarchical Fingertip Space* based stochastic optimization [7], for which an additional constraint is that every grasp candidate is also evaluated for geometric matching. In addition, the hand reachability can be ensured by a grasping manifold [28].

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

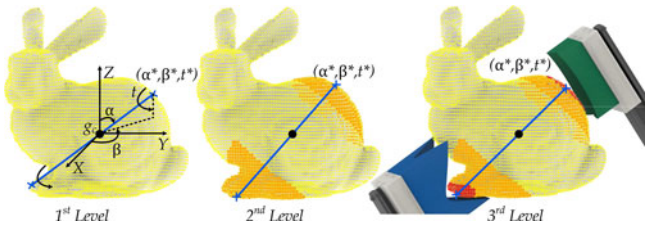

Fig. 5. This figure exemplifies the hierarchical grasp planning on a bunny model using 2 different fingertip designs. In this example, 3 optimization levels have been used, the searched points are rendered in yellow, orange and red for different levels. The grasp planning result is shown on the last level.

In this work, as we will evaluate the proposed approach using the 2-fingered gripper on a Baxter robot, we provide details below for the grasp planner on parallel grippers. Next, we explain how the provided grasp planner addresses the aforementioned 3 considerations:

1) Grasp Stability: The stability of contact-based grasping is usually evaluated based on force closure properties [10]. However, a grasp with 2 contact points can never be force closed. As such, we apply the antipodal grasping as a heuristic to plan stable grasps. Concretely, as there are 2 contact locations for such a grasp, a contact candidate is generated by first generating a grasp center $g_c \in \mathbb{R}^3$ around the geometric center of the object point cloud $\mathcal{P}$. As shown in Fig. 5, a grasp candidate $g = \{(c_1, t_1), (c_2, t_2)\}$ is generated using the ray-shooting method at the 2 utmost intersections on the object mesh model.[1] For taking into account the finger pointing direction, t_i is also generated for each grasp. Since we consider parallel grippers, we have $t_1 \equiv t_2$. In the following, where it does not cause confusions, we will use t to denote the pointing direction of both fingers.

2) Grasp Reachability: To ensure that the generated grasp is reachable by the adopted gripper, let $n_1, n_2 \in \mathbb{S}^2$ be the contact normals at c_1 and c_2, and let $\varepsilon \in \mathbb{R}^+$ be a small value, the following conditions should be satisfied:

$$\begin{aligned} \Omega_1 \leq \|c_1 - c_2\| \leq \Omega_2 \\ \|n_1 + n_2\| \leq \varepsilon \end{aligned} \tag{11}$$

where Ω_1 and Ω_2 denote that minimum and maximum openings of the gripper. The second condition ensures that the contact normals are pointing to each other.

3) Geometric Matching: Given a grasp candidate $g = \{(c_1, t_1), (c_2, t_2)\}$, the corresponding contact areas $\mathcal{C}_1^F, \mathcal{C}_2^F$ are extracted and transformed into the fingertip frame by (1) and (2). To ensure the geometric matching so as to make the maximal contact areas, we aim at solving the optimization problem:

$$(c_1^*, c_2^*, t^*) = \underset{c_1, c_2 \in \mathcal{P}, t \in \mathbb{S}^2}{\arg\min} \; \gamma(\tau(\rho_1), \mathcal{C}_1^F) + \gamma(\tau(\rho_2), \mathcal{C}_2^F) \tag{12}$$

4) Grasp Planning: We solve the grasp planning with the aforementioned considerations using hierarchical optimization [7]. As shown in Fig. 5, the grasp center g_c is randomly sampled in a small range around the geometric center of $\mathcal{P}$. Thereafter, the optimization is carried out in a hierarchical manner. Initially, on the first level of the hierarchical optimization, grasp candidates are generated by ray-shooting to sweep over the spherical coordinate system, centered at g_c, with respect to the angular coordinates $\alpha \in \delta_\alpha$, $\beta \in \delta_\beta$. The finger pointing direction t is swept over the range of δ_t around the z-axis in the fingertip frame. Initially, we set $\delta_\alpha, \delta_\beta = [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\delta_t = [-\pi, \pi]$. In the following, we denote by $g = (\alpha, \beta, t)$ a grasp represented in the spherical coordinates.

[1] pyoctree libarary: https://pypi.python.org/pypi/pyoctree/

Algorithm 2: Hierarchical Grasp Planning for Parallel Grippers.

Input: $\tau(\rho_1), \tau(\rho_2), \mathcal{P}, iter, \kappa, \Omega_1, \Omega_2, \varepsilon$
Output: $g = (\alpha^*, \beta^*, t^*)$

$cost^* \leftarrow \infty, (\alpha^*, \beta^*, t^*) \leftarrow (0, 0, 0)$ ▷ Initialization
for l in $\{1, \ldots, L\}$ **do** ▷ Hierarchy
  $\delta_\alpha \leftarrow [-\frac{\pi}{2\kappa^{l-1}} + \alpha^*, \frac{\pi}{2\kappa^{l-1}} + \alpha^*]$ ▷ Reduce Range
  $\delta_\beta \leftarrow [-\frac{\pi}{2\kappa^{l-1}} + \beta^*, \frac{\pi}{2\kappa^{l-1}} + \beta^*]$
  $\delta_t \leftarrow [-\frac{\pi}{\kappa^{l-1}} + t^*, \frac{\pi}{\kappa^{l-1}} + t^*]$
  for α in $linspace(\delta_\alpha, iter)$ **do**
    for β in $linspace(\delta_\beta, iter)$ **do**
      $(c_1, c_2) \leftarrow getContacts(\mathcal{P}, \alpha, \beta)$
      if not $\Omega_1 \leq \|c_1 - c_2\| \leq \Omega_2 \wedge \|n_1 + n_2\| \leq \varepsilon$ **then**
        Continue ▷(11)
      end if
      for t in $linspace(\delta_t, iter)$ **do**
        $(\mathcal{C}_1^F, \mathcal{C}_2^F) \leftarrow extractAreas(\mathcal{P}, c_1, c_2, t)$
        $cost_1 \leftarrow \gamma(\tau(\rho_1), \mathcal{C}_1^F) + \gamma(\tau(\rho_2), \mathcal{C}_2^F)$
        $cost_2 \leftarrow \gamma(\tau(\rho_1), \mathcal{C}_2^F) + \gamma(\tau(\rho_2), \mathcal{C}_1^F)$
        $cost \leftarrow min(cost_1, cost_2)$ ▷ Finger Swap
        if $cost < cost^*$ **then** ▷(12)
          $(\alpha^*, \beta^*, t^*) \leftarrow (\alpha, \beta, t)$
          $cost^* \leftarrow cost$
        end if
      end for
    end for
  end for
end for

Once the optimal solution $g^* = (\alpha^*, \beta^*, t^*)$ is obtained on the first level by solving (12) constrained by (11), we enter the second level to conduct finer optimization. For this, we introduce a factor $\kappa \in \mathbb{N}$ to reduce the search range to be more focused around g^*. Concretely, around the optimal solution g^* obtained from the first level, the sweeping range is reduced to $\delta_\alpha = [-\frac{\pi}{2\kappa} + \alpha^*, \frac{\pi}{2\kappa} + \alpha^*]$, $\delta_\beta = [-\frac{\pi}{2\kappa} + \beta^*, \frac{\pi}{2\kappa} + \beta^*]$ and $\delta_t = [-\frac{\pi}{\kappa} + t^*, \frac{\pi}{\kappa} + t^*]$.

Following this process, the final optimal solution is obtained at the predetermined bottom level L. Let us denote by $linspace(\delta, iter)$ a function which devides the search range δ into $iter$ search steps, by $(c_1, c_2) \leftarrow getContacts(\mathcal{P}, \alpha, \beta)$ a function that gets the intersections at (α, β) and by $(\mathcal{C}_1^F, \mathcal{C}_2^F) \leftarrow extractAreas(\mathcal{P}, c_1, c_2, t)$ a function that extracts contact areas at (c_1, c_2, t) from $\mathcal{P}$. Given a novel object $\mathcal{P}$, this optimization process is detailed in Algorithm 2 and will be evaluated in Section VI.

VI. Experiments

We evaluate our approach from 3 perspectives: 1) We quantitatively evaluate the contact primitives construction described in Section III-B; 2) We quantitatively evaluate the fingertip designs using the grasp planning algorithm in Section V, and test the grasp stability provided by our fingertip designs against a

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

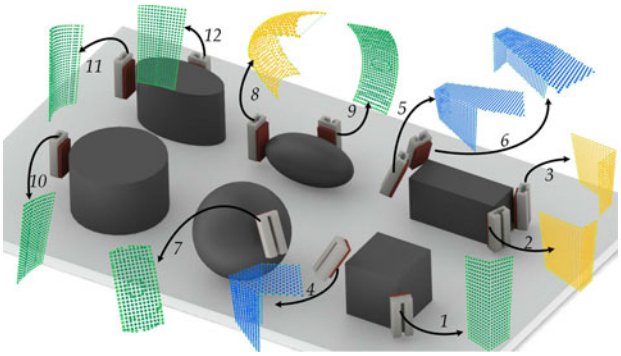

Fig. 6. 12 contact areas are extracted from training objects associated with example grasps. The clustering result is indicated by orange, blue and green colors. The contact areas rendered in each color compose a contact primitive.

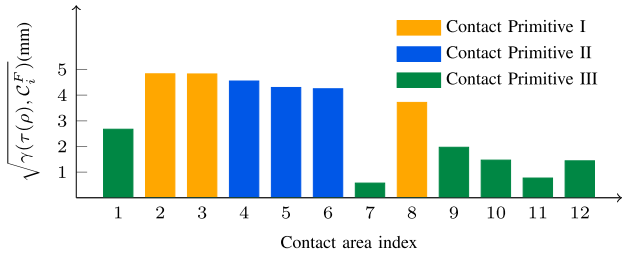

Fig. 7. The differences between all example contact areas and the corresponding fingertip designs. We can see that the differences are very small in comparison to the fingertip's size, this means that the fingertip designs can all well mimic the responsible contact areas in contact primitives.

baseline of normal flat fingertips; and finally 3) We qualitatively show that our design is more robust against uncertainties. In this work, we conduct all experiments using a Baxter robot equipped with a parallel gripper.

A. Contact Primitives and Fingertip Optimization

To construct the contact primitives for guiding our fingertip designs, we adopt a set of 6 training objects, which are simple shape primitives, associated with example grasps generated by the grasp planner in [7] for contact areas extraction. As depicted in Fig. 6, 12 contact areas are extracted to cover a variety of local contact geometries. Note that the extracted contact areas directly determine the fingertip designs and the number of needed fingertips (Section III-B). When different example objects or grasps are used, the fingertips will be designed differently than what will be presented later on.

Once the contact areas are extracted, we cluster them to construct the contact primitives. For this, the number of classes is determined by the Elbow criteria described in Section III-B. Consequently, we cluster the contact areas into 3 contact primitives as shown in Fig. 6, we can see that there are mainly 3 types of local geometries: *edges*, *corners* and *curved surfaces*. The differences between the example contact areas and the corresponding optimized fingertip surfaces are reported in Fig. 7, from which we can see that the difference for each contact area is small, indicating that each fingertip design can well represent all responsible contact areas.

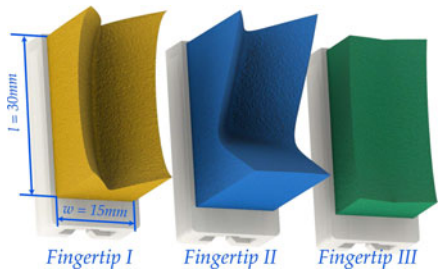

Fig. 8. Fingertip designs for 3 contact primitives, the colors indicate the corresponding contact primitives. Every fingertip is attached to a base on which a customized notch is designed for the Baxter's gripper, so that they can be used by plug-and-play.

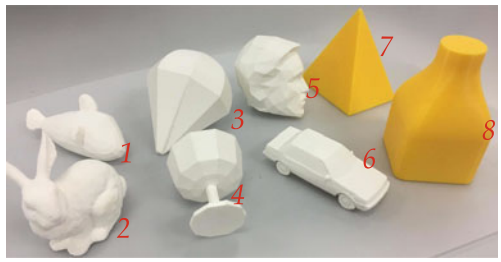

Fig. 9. The 8 3D printed test objects used in this work. The white objects are downloaded from the Princeton Shape Benchmark [29].

Using the fingertip optimization algorithm in Section IV-A, the 3 fingertips are designed in terms of the constructed contact primitives as shown in Fig. 8.

B. Grasp Planning on Novel Objects

Having designed the fingertips, we evaluate the designs based on a set of novel objects shown in Fig. 9. As we have designed 3 fingertips while only 2 of them can be installed on Baxter at the same time, we test them by different pairs of 2 fingertips: I and I, I and II, as well as II and III. As a baseline comparison, we will include a pair of normal flat fingertips in all experiments below.

1) Optimization Residual: For grasping the novel objects, the grasps are planned using the hierarchical algorithm described in Section V in terms of the chosen fingertip pair. Some of the grasps planned by Algorithm 2 using our fingertip designs are shown in Fig. 10. Firstly, the optimization residuals of Algorithm 2 are reported in Fig. 11. In comparison to Fig. 7, we can observe that the planned grasp contacts possess local geometries that are very similar to the designed fingertips, independent of which fingertip pair is chosen.

As such, it has further confirmed our motivation that most grasp contacts on daily objects share just a few contact local geometries. This fact can also be intuitively understood, since all daily objects possess closed shapes, i.e., they do not extend to infinity and their scales are limited. There must exist local geometries such as *edges*, *corners* and *curved surfaces* at many locations on object surfaces. Therefore, it is straightforward to understand that, given the designed 3 fingertips, we can almost grasp any shapes as long as the object scale fits into the gripper's opening range.

Moreover, it is worthwhile to notice that the optimization residuals for flat fingertips are much lower than those of our

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

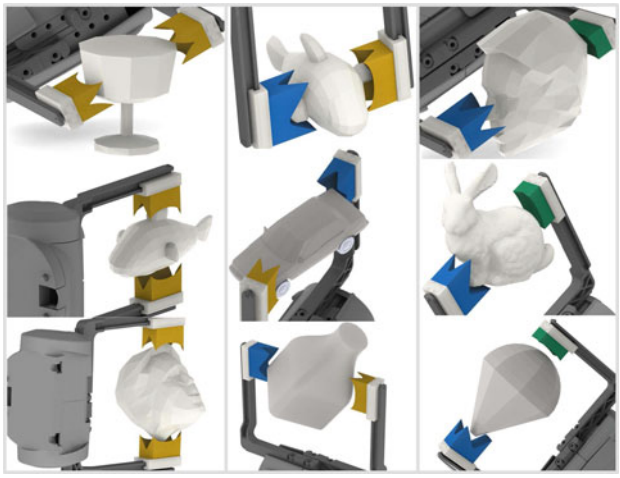

Fig. 10. Example grasp planning results on novel objects.

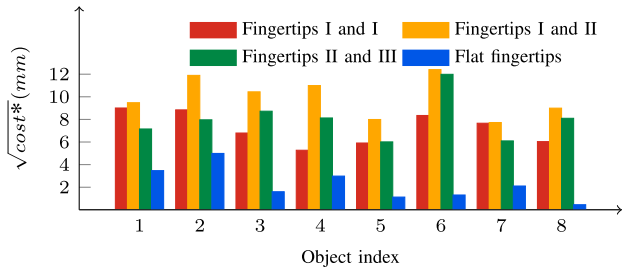

Fig. 11. Grasp planning optimization residuals of Algorithm 2. We can observe that the residuals are comparable to the results in Fig. 7, indicating that our grasp planner can find stable grasp contacts that match the fingertip designs, and that most grasp contacts share just a few classes of local geometries.

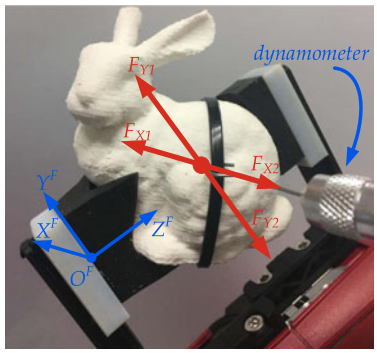

Fig. 12. The grasp stability is measured by a dynamometer. Concretely, we pull the object through the center of mass in both positive and negative directions of axes X^F and Z^F, the minimal force which breaks the grasp in one of the 4 directions is recorded as the stability performance. In case a grasp cannot be broken from any directions up to the dynamometer's limit (30N), we record 30N as the result.

fingertip designs. This is easy to understand as the test objects all have low-curvature areas on their surfaces. However, as will be shown in the following experiments, grasping at flat areas is much weaker in withstanding external forces.

2) Stability Test: Furthermore, we have evaluated the stability of each planned grasp on the test objects. The experiment design is explained in Fig. 12. In this test, we have 3D printed

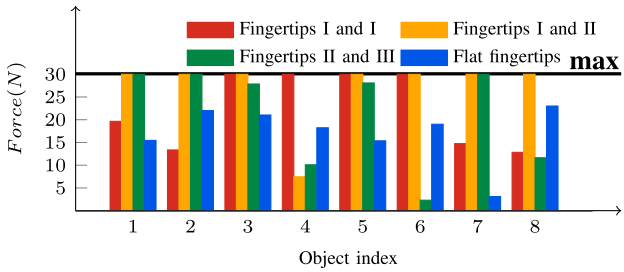

Fig. 13. Stability test results. In this plot, the black line indicates the upper limit of our dynamometer.

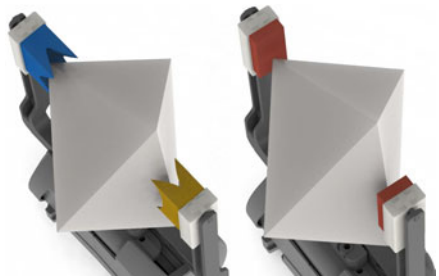

Fig. 14. An example showing the comparison between our fingertip designs and the normal flat fingertips.

the defingertips using soft materials[2] to compensate for the optimization residuals reported in Fig. 11. To ensure fair comparisons, we 3D printed the flat fingertips using the same material. In this experiment, all the grasps are planned by Algorithm 2, including the ones using the flat fingertips.

The stability test results are reported in Fig. 13. In most of the cases, we can see that our fingertip designs significantly outperformed the flat fingertips. In some other cases, the flat fingertips performed better, since there are certain objects not suitable for specific fingertip pairs. For example, the object 6 which is a car model cannot be well grasped by a pair of fingertips II and III, as it is not possible to find an antipodal grasp composed by a *corner* contact and a *curved surface* contact.

Importantly, we can observe not only that our designs can significantly improve the grasp stability, but also that our designs in many cases can provide grasps that exceed the dynamometer's upper limit, which is larger than the payload of the Baxter robot (2.2 kg). Although we do not provide proofs in this work, this is because the grasps using our fingertip designs actually in many cases form caging grasps [17], which can never be broken unless breaking the fingertips.

C. Uncertainties

Positioning errors cannot be fully neglected due to errors in controller, perception, etc. For contact-based grasping, positioning errors are fatal for grasp executions [11]. As shown in Fig. 14, when executing a grasp using flat fingertips with contacts at non-flat areas, a little positioning error can easily cause a grasp to fail. However, as can be observed, using our fingertip designs can tolerate the positioning errors to a much larger extent, since the object is more constrained and the errors can be corrected by compliant motions.

[2]TangoBlackPlus: http://www.stratasys.com/materials/polyjet/rubber-like.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.

Additionally, when the grasp contacts are made at areas that match the fingertip geometries, our designs exert forces from a range of positions and directions around the contact center. This further makes the grasps insensitive to friction changes and perception errors, and allows grasp executions in more uncertain environments.

VII. Conclusion

In this work, we have addressed the problem of fingertip surface design by leveraging on the fact that most grasp contacts on daily objects share just a few classes of contact local geometries. As such, for fingertip optimization, we can first learn the most representative contact geometries from a set of grasps on example objects, and then optimize the fingertip surfaces to maximally mimic the shapes of the learned contact geometries. Using the designed fingertips, we can on an object surface find grasp contacts that can maximally mimic the shapes of the fingertips, so as to improve grasp stability and robustness by maximizing grasp contact areas.

Formally, we have defined the concept of *Contact Primitive*, which is a set of contact areas, to represent a class of similar local geometries. Thereafter, by optimally clustering a set of example contact areas extracted from example grasps, we constructed a finite set of *Contact Primitives*, which has the minimum sum of in-class differences, to represent a finite set of contact geometries. For fingertip optimization, we model the fingertip surface by Bézier surface, and then optimize the control points to minimize the differences between the fingertip surface and contact areas in the corresponding contact primitive. For grasping novel objects, we provided a hierarchical grasp planner to find grasp contacts that can maximize contact areas with the designed fingertips, while forming a stable and reachable grasp.

The proposed approach has been evaluated on a Baxter robot with parallel grippers. According to the example objects associated with example grasps, we designed 3 fingertips in total to cover contact geometries of *corners*, *edges* or *curved surfaces*. The evaluation results have shown that our designs: 1) can well represent the grasp contacts on example objects; 2) work well for many other novel daily objects; 3) significantly improve grasp stability in comparison to a baseline of flat fingertips; and 4) are more robust against uncertainties.

In the future work, we plan to apply our approach on multi-fingered hands, which can be equipped with more combinations of fingertip designs, to avoid the problem of not finding a good grasp on a certain object due to improper fingertip combinations. Furthermore, we would like to study the topological properties of our fingertip designs, in order to plan caging grasps for even better grasp robustness.

References

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 348–353.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.

[3] A. T. Miller and P. K. Allen, "Graspit! A versatile simulator for robotic grasping," *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[4] Y. Zheng, M. C. Lin, and D. Manocha, "On computing reliable optimal grasping forces," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 619–633, Jun. 2012.

[5] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 2290–2295.

[6] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2002, pp. 1553–1559.

[7] K. Hang *et al.*, "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 960–972, Aug. 2016.

[8] B. Sundaralingam and T. Hermans, "Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization," in *Robot.: Sci. Syst.*, 2017.

[9] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2003, vol. 4, pp. 3692–3697.

[10] M. Roa and R. Suárez, "Grasp quality measures: Review and performance," *Auton. Robot.*, vol. 38, no. 1, pp. 65–88, 2015.

[11] M. A. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-D objects," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 839–850, Aug. 2009.

[12] K. Hang, F. T. Pokorny, and D. Kragic, "Friction coefficients and grasp synthesis," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013, pp. 3520–3526.

[13] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robot. Auton. Syst.*, vol. 75, pp. 352–364, 2016.

[14] M. Honarpardaz *et al.*, "Generic automated multi-function finger design," *IOP Conf. Ser, Mater. Sci. Eng.*, vol. 157, no. 1, 2016.

[15] M. Honarpardaz, M. Tarkian, J. lvander, and X. Feng, "Finger design automation for industrial robot grippers: A review," *Robot. Auton. Syst.*, vol. 87, pp. 104–119, 2017.

[16] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 616–629, Jun. 2011.

[17] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," in *Proc. Robot., Sci. Syst.*, 2011.

[18] C. D. Santina, G. Grioli, M. Catalano, A. Brando, and A. Bicchi, "Dexterity augmentation on a synergistic hand: The pisa/iit softhand+," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2015, pp. 497–503.

[19] H. Dai, A. Majumdar, and R. Tedrake, "Synthesis and optimization of force closure grasps via sequential semidefinite programming," in *Int. Symp. Robot. Res.*, 2015, pp. 285–305.

[20] Y.-H. Liu, M.-L. Lam, and D. Ding, "A complete and efficient algorithm for searching 3-D form-closure grasps in the discrete domain," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 805–816, Oct. 2004.

[21] H. Dang and P. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 1311–1317.

[22] J. Mahler *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot, Sci. Syst.*, 2017.

[23] K. Hang, J. A. Stork, and D. Kragic, "Hierarchical fingertip space for multi-fingered precision grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014, pp. 1641–1648.

[24] K. Hang, J. A. Stork, N. S. Pollard, and D. Kragic, "A framework for optimal grasp contact planning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 704–711, Apr. 2017.

[25] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: An analysis and critique," *Strateg. Manage. J.*, vol. 17, no. 6, 1996, Art. no. 441458.

[26] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann, 2002.

[27] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*. New York, NY, USA: Springer-Verlag, 2006.

[28] K. Hang, J. A. Haustein, M. Li, A. Billard, C. Smith, and D. Kragic, "On the evolution of fingertip grasping manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 2022–2029.

[29] M. K. Philip Shilane, P. Min, and T. Funkhouser, "The princeton shape benchmark," in *Proc. Shape Model. Int.*, 2004, pp. 167–178.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on March 11,2021 at 06:18:57 UTC from IEEE Xplore. Restrictions apply.