Privacy-Preserving Cloud-Based Grasp Planning

Jeffrey Mahler[1,*], Brian Hou[1,*], Sherdil Niyaz[1], Florian T. Pokorny[1], Ramu Chandra[3], Ken Goldberg[1,2]

***Abstract*—To support industrial automation, systems such as Grasp-it! and Dex-Net 1.0 provide Grasp Planning as a Service (GPaaS). For example, a manufacturer setting up an automated assembly line for a new product can upload part geometry to the service and receive a ranked set of robust grasp configurations, and the GPaaS can accelerate future grasp planning by statistically analyzing grasps on the part. However, many industrial users may be reluctant to share proprietary details of product geometry with any outside parties. This paper defines a privacy-preserving approach to grasp planning and presents an algorithm where a masked version of the part boundary is uploaded along with stable pose configurations, allowing proprietary aspects of the part boundary to remain confidential. One challenge is the tradeoff between grasp coverage and privacy: balancing the desire for a rich set of alternative grasps (coverage) based on analysis of graspable surfaces against the user's desire to maximize privacy. We introduce a grasp coverage metric based on dispersion, a coverage metric used in motion planning, and we formalize its relationship with privacy (the amount of the object surface that is masked). We implement our algorithm for Dex-Net 1.0 and present case studies of the privacy-coverage tradeoff on a set of 23 industrial parts. Our results suggest that masking the part using the convex hull of the proprietary zone prunes grasps in collision and provides grasp coverage with low distortion of the object similarity metric used to accelerate grasp planning in Dex-Net 1.0. We also find empirically that increasing privacy always leads to decreasing coverage, and that coverage of the set of all grasps with non-zero robustness decreases with an increasing robustness threshold.**

I. Introduction

Rather than performing computation in isolation, Cloud-based Robotics and Automation systems utilize centrally-hosted computational and storage resources, planning actions based on shared libraries of product data, prior sensor readings, and maps [17]. Recent research suggests that systems providing Grasp Planning as a Service (GPaaS) such as GraspIt! [6] and Dex-Net 1.0 [26] can reduce the time required to plan a diverse set of robust grasps to cover a new object by leveraging prior 3D object models labeled with robotic grasps and grasp quality metrics. The speedup may increase with the amount of prior models and grasps, motivating the development of Cloud-based shared and growing datasets where users can upload new part geometry to a GPaaS and receive a ranked set of robust grasp configurations. A Cloud-based GPaaS also eliminates the need for platform-specific software updates and maintenance for individual users.

One problem for Cloud-based planners is that proprietary 3D geometric data such as connectors between parts, the diameter of turbine shafts, or gear ratios and pitches can be compromised via two attacks: (a) intercepting a part in transmission to the Cloud or (b) querying the part from a shared dataset. This may allow competitors to acquire design parameters that may have been the result of complex and costly simulations, motivating methods to mask proprietary sections of a part boundary before transmission. However, planning on a masked part may reduce the number of grasps returned by the GPaaS and, in extreme cases, may prevent a planner from finding any grasps for an object. This raises the question: how can we balance the desire for a rich set of grasps covering the part surface with the user's desire to maximize privacy?

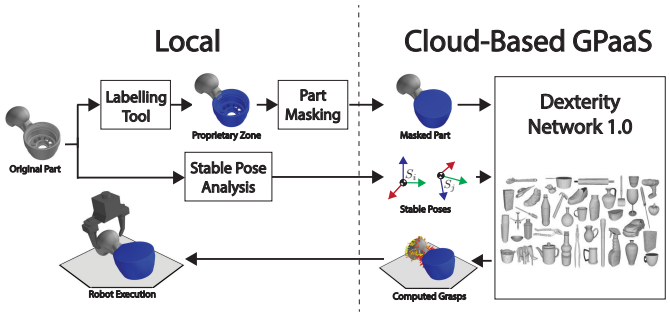

Fig. 1: Overview of our proposed methodology for privacy-preverving grasp planning. Industrial users label proprietary zone of the part with a graphical interface. The masked object is then transmitted to a Cloud-based grasp planner along with its stable poses. The grasp planner computes a set of grasps for each stable pose and returns the grasp sets to the user.

In this paper we introduce the problem of privacy-preserving grasp planning: to plan a set of robust grasps on a masked part boundary that best covers the surface of the unmasked object. We define a grasp coverage metric based on dispersion, a metric of sample coverage used in motion planning [23], and define part privacy based on the percentage of the mesh surface that is masked. We present an algorithm for planning a covering set of robust and collision-free grasps on a masked part given a set of stable poses for the part on a planar worksurface and the geometry of a parallel-jaw gripper. We formalize the privacy-coverage tradeoff for our algorithm, showing that coverage cannot increase as the part becomes more private.

We implement our algorithm in Dex-Net 1.0 [26] with tools for labeling proprietary zones of parts and analyzing object stable poses and inertial properties before transmitting the data to a GPaaS as illustrated in Fig. 1. We study the privacy-coverage tradeoff and the tradeoff between coverage and the robustness of planned grasps for a set of 23 parts. We compare three part masking methods: removing the proprietary zone on the mesh, replacing each connected component of the proprietary zone with a bounding box, and replacing each connected component of the proprietary zone with its

* These authors contributed equally to the paper
[1]Department of Electrical Engineering and Computer Sciences; {jmahler, brian.hou, sniyaz, ftpokorny, goldberg}@berkeley.edu
[1-2]University of California, Berkeley, USA
[3]Siemens, Berkeley, USA ramu.chandra@siemens.com

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

convex hull. Our experiments suggest that using only the non-proprietary zone in planning may lead to grasps that are in collision on the true object, and that masking the proprietary zone using the convex hull provides lower dispersion and lower distortion of the object similarity metric from Dex-Net 1.0 than bounding-box masking. Furthermore, experiments suggest that coverage does not increase with increasing privacy or robustness.

II. Related Work

The goal of grasp planning is to find a set of grasps for an object that optimizes a grasp quality metric [11], [35] or the number of successes in physical trials when the object and contact locations are known exactly. However, in practice these are not known precisely due to imprecision in perception and control. Several methods have been developed to handle uncertainty in object pose [33] or contact location [44], but these methods cannot be easily extended to handle multiple sources of uncertainty. Robust grasp planning handles uncertainty in multiple quantities by finding a set of grasps that maximize an expected quality metric under a set of sampled perturbations in quantities such as object shape [15], [25], object pose [41], and robot control or friction [22], [26].

Robust grasp planning may be computationally demanding when the space of uncertain quantities is high dimensional. Thus, recent research has studied precomputing a set of grasps for an object offline and storing robust grasps in a database. Weisz et al. [41] computed the probability of force closure P_F under object pose uncertainty for a subset of grasps in the Columbia grasp database [12] and showed that P_F was better correlated with physical grasp success than deterministic metrics. Brook et al. [5] developed a model to predict physical grasp success based on a set of robust grasps planned on a database on 892 point clouds. Kehoe et al. [16] transferred grasps evaluated by P_F on 100 objects in a Cloud-hosted database to a physical robot by retrieving the objects with the Google Goggles object recognition engine. Recently, Mahler et al. [26] created the Dexterity-Network (Dex-Net) 1.0, a dataset of over 10,000 objects and 2.5 million grasps, each labelled with P_F under uncertainty in object shape, pose, and gripper positioning, and used the dataset to speed up planning of a single robust grasp. In comparison, we present an algorithm that plans a covering set of grasps to ensure reachability under different accessbility conditions subject to preserving proprietary part geometry. Other recent research has used databases of 3D models [14] or images [34] to directly regress to the probability of grasp success from simulation or physical trials.

Cloud-based grasp planners raise the issue of how to store and transmit data without compromising proprietary geometric information [36]. This is an example of "privacy over structured data," a common topic in database research in which deterministic techniques are used to preserve privacy for widely-used data analytics [4]. In robotics and automation systems, security is a major topic of interest for the smart grid [19] and manufacturing pipelines [13], and has also been studied in the context of hijacking unmanned aerial vehicles (UAVs) [18] and ground vehicles [40]. Our methods are closely related to past work on the security of 3D models. Early research considered schemes that embed information such as the model owner directly into the geometry to identify theft, for example by using the spectral domain of the mesh [31]. Koller et al. [20] developed a rendering system that allows users to view low-resolution copies of the entire model and request high-resolution snippets from a protected server to prevent acquisition of the entire model geometry. In industry models are often protected using industrial computer-aided design (CAD) software, which is usually bundled with tools for removing details from a model. Solidworks [3] and Autodesk Inventor [1] both contain tools for "defeaturing" a mesh by filling holes, smoothing details, and removing internal features. Other techniques include low-pass filtering [37], Finite Element Re-meshing [29], and feature suppression [10].

Our notion of grasp coverage is also closely related to past research in motion planning and grasping. In motion planning, Lavalle et al. [23] introduced the notion of dispersion to construct deterministic sampling strategies for Probabilistic Roadmap Planners that better cover the configuration space. [23]. This research has been extended to adaptive sampling strategies that reduce dispersion [24] and to deterministic sampling strategies for $SO(3)$ by Yershova et al. [43]. In grasping, coverage research has focused on sampling dense grasp and motion sets for finding grasps in cluttered scenes [9], adaptive sampling of robust grasps over an object surface [7], or analyzing the space of all possible grasps on polygonal objects [39]. However, formal methods for measuring the coverage of grasp sets are relatively less studied. In this work, we introduce a formal notion of grasp coverage based on the dispersion between the set of planned grasps and all possible grasps on the object.

III. Definitions and Problem Statement

In this paper, we consider the pre-computation of a set of robust parallel-jaw grasps for a 3D object model using a masked version that obscures proprietary geometric information such as part connectors. Our goal is to plan a set of grasps Γ on the masked object such that the computed grasp set is robust and covers the non-proprietary surface of the original object.

A. Assumptions

We assume a given binary quality metric $S(\mathbf{g})$ that maps grasps to $\{0, 1\}$ and measure grasp quality by robustness, or the probability of success $P_S(\mathbf{g}) = \mathbb{E}[S(\mathbf{g})]$ under uncertainty due to imprecision in sensing and control. In this paper, we use the probability of force closure P_F under uncertainty in object pose, gripper pose, and friction coefficient as the quality metric, soft finger point contacts, and a Coulomb friction model. For more details on our uncertainty and force closure model, see [26]. We assume the exact object shape is given as a compact surface in units of meters with a given center of mass $\mathbf{z} \in \mathbb{R}^3$.

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

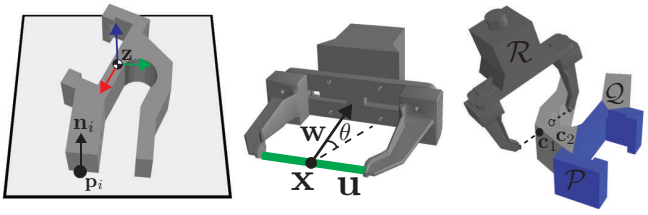

Fig. 2: Illustration of our models for objects and grasps. (Left) The object frame of reference is centered at the center-of-mass $\mathbf{z}$. Each object is associated with a set of stable poses (planar worksurface orienations) defined by the plane $(\mathbf{n}_i, \mathbf{p}_i)$. (Middle) We can parameterize parallel-jaw grasps by their center $\mathbf{x}$ and axis $\mathbf{u}$, which defines a gripper pose when the angle θ of the gripper approach axis $\mathbf{w}$ is specified. (Right) A parallel-jaw gripper $\mathcal{R}$ contacts a mesh $\mathcal{M}$ at points $\mathbf{c}_1$ and $\mathbf{c}_2$. The space of all possible grasps is the space of all contact pairs. Each mesh is divided into a private region $\mathcal{P}$ (blue) and public region $\mathcal{Q}$ (grey).

B. Object Parameterization

We use the object parameterization illustrated in Fig. 2. We parameterize each object as a mesh $\mathcal{M} \subset \mathbb{R}^3$. We represent a mesh $\mathcal{M}$ as the tuple $(\mathcal{V}, \mathcal{T})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{T}$ is a set of triangles interpolating 2-dimensional surfaces between the vertices. Each vertex $\mathbf{v} \in \mathcal{V}$ is specified as a point in 3D space and each triangle $\mathbf{t} \in \mathcal{T}$ is specified as a triplet of vertex indices. All vertices of $\mathcal{M}$ are specified with respect to a reference frame centered at the object center of mass $\mathbf{z}$ and oriented along the principal axes of the vertex set.

We model the object as resting on an infinite planar worksurface under quasi-static conditions with a uniform prior distribution on part orientation. Under this assumption the object rests in a stable pose, or orientation such that the object remains in static equlibrium on the worksurface [28], [42]. A triangular mesh has a finite set of stable poses $\mathcal{S} = \{S_1 ..., S_\ell\}$ modulo rotations about an axis perpendicular to the worksurface, and each stable pose S_i is parameterized by the table normal $\mathbf{n}_i$ and a point on the object touching the table surface $\mathbf{p}_i$.

C. Object Privacy

We assume that attackers are third parties that either (a) intercept the part while it is being transmitted to a public Cloud-Based GPaaS or (b) recover the part from queries to a public Cloud-Based GPaaS. Our goal is to hide the exact geometry of each part from attackers, which may be optimized for gas flows, mechanical efficiency, or faster assembly.

To protect privacy, let each object $\mathcal{M} = (\mathcal{V}, \mathcal{T})$ be equipped with a privacy mask, or function $Z : \mathcal{T} \to \{0, 1\}$ such that a triangle $\mathbf{t} \in \mathcal{T}$ must remain private if $Z(\mathbf{t}) = 1$. We denote by $\mathcal{P}(\mathcal{M}, Z) = \{\mathbf{t} \in \mathcal{T} \mid Z(\mathbf{t}) = 1\}$ the private region of the object and $\mathcal{Q}(\mathcal{M}, Z) = \mathcal{M} \setminus \mathcal{P}(\mathcal{M}, Z)$ the public region. We create a masked version of the object $\varphi_Z(\mathcal{M})$ using a *masking function* φ_Z such that $\varphi_Z(\mathcal{P}) \neq \mathcal{P}$ and $\varphi_Z(\mathcal{Q}) = \mathcal{Q}$. We measure the degree of privacy for a mesh by γ, the ratio of the surface area of $\mathcal{P}$ to the total surface area:

$$\gamma(\mathcal{M}, Z) = \text{Area}(\mathcal{P}(\mathcal{M}, Z))/\text{Area}(\mathcal{M}).$$

D. Grasp Parameterization

Our grasp parameterization is illustrated on the right side of Fig. 2. Given an object $\mathcal{M}$, let $\mathcal{G}(\mathcal{M}) = \mathcal{M} \times \mathcal{M}$ be the space of all possible contact point pairs on the object, and let $\mathbf{g} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathcal{G}$ be a parallel-jaw grasp. We can alternatively describe a grasp $\mathbf{g}$ by the midpoint of the jaws in 3D space $\mathbf{x} \in \mathbb{R}^3$ and approach axis $\mathbf{u} \in \mathbb{S}^2$ where

$$\mathbf{x} = \frac{1}{2}(\mathbf{c}_1 + \mathbf{c}_2) \qquad \text{and} \qquad \mathbf{u} = \frac{\mathbf{c}_2 - \mathbf{c}_1}{\|\mathbf{c}_2 - \mathbf{c}_1\|_2}.$$

We can also convert a grasp $\mathbf{g}$ to a gripper pose $T(\mathbf{g}, \theta) \in SE(3)$ relative to the object by specifying an angle θ of the gripper approach axis $\mathbf{w}$.

E. Grasp Subsets

Let $\mathcal{R}$ denote a mesh model of a robot gripper and $\mathcal{R}(\mathbf{g}, \theta)$ denote the gripper model in pose $T(\mathbf{g}, \theta)$. Of particular interest are the following subsets of grasps:

Reachable Grasp Set, $\mathcal{X}(\mathcal{R}, S_i)$: The reachable grasp set is the set of grasps on $\mathcal{M}$ such that $\mathcal{R}(\mathbf{g}, \theta)$ does not collide with the object $\mathcal{M}$ or the worksurface for stable pose S_i.

Robust Grasp Set, $\mathcal{Y}(\tau)$: The set of grasps on $\mathcal{M}$ with P_S greater than some threshold τ.

Executable Grasp Set, $\mathcal{E}(\mathcal{R}, S_i, \tau)$: The intersection of the reachable and robust grasp sets: $\mathcal{E} = \mathcal{X} \cap \mathcal{Y}$.

F. Grasp Coverage

Consider an arbitrary grasp set $\Upsilon \subseteq \mathcal{G}$ on object $\mathcal{M}$ and a discrete set of planned grasps $\Gamma \subset \Upsilon$. We measure the extent to which Γ covers Υ using dispersion [23], [30], a measure of coverage previously used to analyze sampling-based motion planners.

To measure coverage, we first need a notion of grasp distance. We measure the distance between grasps for object $\mathcal{M}$ by a function $\rho : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ [21], where:

$$\rho(\mathbf{g}_i, \mathbf{g}_j) = \lambda(\mathcal{M})\|\mathbf{x}_i - \mathbf{x}_j\|_2 + (2/\pi)\arccos(|\langle \mathbf{u}_i, \mathbf{u}_j \rangle|)$$

where $\lambda(\mathcal{M})$ is a constant controlling the relative weighting of the distance between the grasp center and axis. In this work we choose $\lambda(\mathcal{M})^{-1} = \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{V}} \|\mathbf{x}_i - \mathbf{x}_j\|_2$ to put equal weighting between the center and axis distances.

Dispersion, illustrated in Fig. 3, is formally defined as [24]:

$$\delta(\Gamma, \Upsilon) = \sup_{\mathbf{g}_j \in \Upsilon} \min_{\mathbf{g}_i \in \Gamma} \rho(\mathbf{g}_i, \mathbf{g}_j).$$

In the case of $\Gamma = \varnothing$, we let $\delta(\Gamma, \Upsilon) = \infty$. Intuitively, δ measures the radius of the largest ball (under ρ) in Υ that does not touch any samples in Γ.

Definition III.1. *The coverage for Γ with respect to Υ is* $\alpha(\Gamma, \Upsilon) = \exp(-\delta(\Gamma, \Upsilon))$.

Coverage approaches 1 as dispersion decreases and is approximately zero as the dispersion becomes infinite.

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

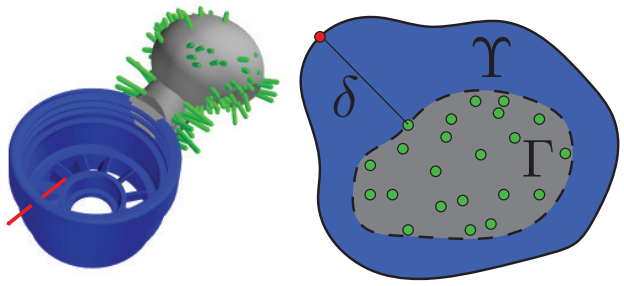

Fig. 3: Illustration of the grasp dispersion metric δ. (Left) In the workspace, the public region of an object (grey) is covered by a set of grasps Γ (green). Each grasp is illustrated by a line segment with orientation $\mathbf{v}$ centered at $\mathbf{x}$. Each grasp is a sample from a larger space of possible grasps Υ, such as the set of all possible grasps on the part. The farthest grasp in Υ from grasps in Γ is shown in red. (Right) We measure coverage by the dispersion δ, or the radius of the largest empty ball centered in Υ. Lower dispersion indicates higher grasp coverage.

G. Objective

Our formal objective is to plan a set of n grasps $\Gamma = \{\mathbf{g}_1, ..., \mathbf{g}_n\}$ on the masked object such that $\Gamma \subset \mathcal{E}(\mathcal{R}, S_i, \tau)$ and the coverage $\alpha(\Gamma, \mathcal{E})$ is as small as possible. Note that Γ must be a subset of the grasp sets on the original object, even though it is planned using the masked version.

IV. Privacy-Preserving Grasp Planning Algorithm

Algorithm 1 details our algorithm for privacy-preserving grasp planning, which is also illustrated in Fig. 1. The algorithm takes as input the object mesh $\mathcal{M}$, a masking function φ (see Section V), and parameters for the executable grasp set, and returns a set of grasps Γ_i and robustness metrics R_i for each stable pose S_i of the object. We mask the object and compute stable poses before transmission, then compute a set of candidate public grasps by considering all possible pairs of contacts at mesh triangle centers, and then prune grasps based on collisions and robustenss to form a subset of the executable grasp set for each stable pose. We measure the robustness P_S of each grasp using the probability of force closure P_F under object pose, gripper pose, and friction uncertainty, and compute P_F using Monte-Carlo integration (for more details, see [26]).

A. Grasp Candidate Generation

We form a set of candidate grasps for each object by forming a set of candidate contact points from the mesh triangle centers and then evaluating and pruning pairs of possible contacts. In order to ensure that the set of contacts covers the mesh surface, we first subdivide triangles of the masked mesh using primal triangular quadrisection [32] until the maximum edge length of each triangle is less than some threshold ϵ, transferring the privacy label $Z(\mathbf{t}_i)$ from each triangle to its children. We then use the set of triangle centers on the public zone of the subdivided mesh as our set of candidate contacts $\mathcal{C}$ since the geometry of triangles in the proprietary zone may have been altered. The triangle subdivision step increases the density of our candidate grasp set.

B. Privacy-Coverage Tradeoff

The set of possible contacts decreases as the surface of the part becomes more private, , which intuitively would to a smaller grasp set and therefore smaller coverage. This property holds formally for the Privacy-Preserving Grasp Planning Algorithm. Consider a part with two masks Z_1 and Z_2 such that proprietary zones are nested, $\mathcal{P}(\mathcal{M}, Z_1) \subset \mathcal{P}(\mathcal{M}, Z_2)$. Then the candidate grasp sets $\mathcal{G}_1$ and $\mathcal{G}_2$ are also nested, $\mathcal{G}_2 \subset \mathcal{G}_1$. If $n > |\mathcal{G}_i|$ then the loop on line 15 terminates only once all possible contact pairs have been evaluated, and thus the planned grasp sets are also nested $\Gamma_2 \subseteq \Gamma_1$. Therefore $\alpha(\Gamma_1, \mathcal{E}) \geqslant \alpha(\Gamma_2, \mathcal{E})$.

1 **Input:** Object Mesh $\mathcal{M}$, Masking Function φ, Robot Gripper $\mathcal{R}$, Quality Threshold τ, Stable Pose Threshold p, Number of Grasps n, Edge Length Threshold ϵ, Robustness metric P_S
Result: Grasp Set Γ and Robustness Metrics R
`// Mask mesh and analyze stable poses`
2 $\mathcal{S} =$ StablePoses($\mathcal{M}, p$);
3 $Z =$ UserLabel($\mathcal{M}$);
4 $\varphi_Z(\mathcal{M}) =$ Mask($\mathcal{M}, Z, \varphi$);
`// Generate grasp candidates`
5 $\varphi_Z(\mathcal{M}) =$Subdivide($\varphi_Z(\mathcal{M}), \epsilon$);
6 $\mathcal{C} = \Gamma = R = \varnothing$;
7 **for** $\mathbf{t} \in \varphi_Z(\mathcal{T})$ **do**
8 **if** $Z(\mathbf{t}) = 0$ **then**
9 $\mathcal{C} = \mathcal{C} \cup \{$Center($\mathbf{t}$)$\}$;
10 **end**
11 **end**
`// Compute cover for each stable pose`
12 **for** $S_i \in \mathcal{S}$ **do**
13 $\Gamma_i = \varnothing, R_i = \varnothing, j = 0$;
14 $\mathcal{G}_i =$ Shuffle($\mathcal{C} \times \mathcal{C}$);
15 **while** $|\Gamma_i| < n$ *and* $j < |\mathcal{G}_i|$ **do**
16 $\mathbf{g} = \mathcal{G}_i[j]$;
17 **if** $\mathbf{g} \notin \Gamma_i$ *and* $P_S(\mathbf{g}) > \tau$ *and* *NoCollision*($\mathbf{g}, S_i, \mathcal{R}, \varphi_Z(\mathcal{M})$) **then**
18 $\Gamma_i = \Gamma_i \cup \{\mathbf{g}\}$, $R_i = R_i \cup \{P_S(\mathbf{g})\}$;
19 **end**
20 $j = j + 1$;
21 **end**
22 $\Gamma = \Gamma \cup \{\Gamma_i\}, R = R \cup \{R_i\}$;
23 **end**
24 return Γ, R;

Algorithm 1. Privacy-Preserving Grasp Planning

V. Part Masking

Before transmitting the part across a network for grasp planning, the part must be masked to ensure that proprietary geometry is not compromised. Our proposed method, illustrated in the left panel of Fig. 1, consists of a labeling tool for industrial users to select proprietary zones via a graphical user interface and a mask application stage before transmission.

A. Labeling Tool

To use our graphical tool for labeling the proprietary zones of parts, a user first loads a mesh and orients the mesh such that the proprietary zone of the mesh lies within a bounding box in a graphical user interface. Then the user drags the mouse to form a box in pixel coordinates, and any triangles that project

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

within the bounding box are labeled private. The labeled region of the part is then colored blue for the user to either accept or reject the label. If the label is accepted then we save a binary label for each triangle $Z(\mathbf{t}_i)$ such that $Z(\mathbf{t}_i) = 1$ if triangle $\mathbf{t}_i$ is private and $Z(\mathbf{t}_i) = 0$ if not.

B. Masking Methods

Fig. 4 illustrates the three methods we compare for obscuring the geometry of a part with a mask. Each method produces a masked part $\varphi_Z(\mathcal{M}) = (\varphi_Z(\mathcal{V}), \varphi_Z(\mathcal{T}))$ from the original part $\mathcal{M} = (\mathcal{V}, \mathcal{T})$.

Deleted Mesh. The masked triangle list $\varphi_Z(\mathcal{T})$ contains all triangles from the public zone of the mesh ($Z(\mathbf{t}_i) = 0$) and all triangles from the private zone ($Z(\mathbf{t}_i) = 1$) are deleted. The masked vertex list $\varphi_Z(\mathcal{V})$ contains all vertices that are referenced by a triangle in $\varphi_Z(\mathcal{T})$. One potential shortcoming of this method is that some areas on the masked object may appear reachable by a gripper but cannot be reached on the true object due to collisions.

Bounding Box. The masked part $\varphi_Z(\mathcal{M})$ contains all triangles and vertices from the public zone of the mesh, and triangles and vertices from the private zone are broken into connected components. Each connected component is replaced by a cube oriented along the rotational axes of the reference frame for the original part. The bounding boxes are zippered to the original mesh [27], [38]. This method preserves the reachable areas of the part, however the size of the bounding boxes can prune grasps that are reachable on the original part.

Convex Hull. The masked part $\varphi_Z(\mathcal{M})$ contains all triangles and vertices from the public zone of the mesh. Triangles and vertices from the private zone are broken into connected components, each of which is replaced by its convex hull. The convex hulls are zippered to the orignal mesh [27], [38]. This method preserves the reachable areas of the part but may also induce collisions for grasps that are collision-free on the original part.

VI. Experiments

We implemented the described algorithm for privacy-preserving grasp planning in Dex-Net 1.0 and planned grasp sets $\Gamma \subset \mathcal{E}$ for a set of 23 parts from Thingiverse [2]. Unless otherwise noted, our experiments used a number of grasps $n = 10,000$, a P_F threshold of $\tau = 0.01$, and an edge length threshold of $2.0cm$. We compute the stable poses for each object following [42] and use the stable pose with highest probability of occurence under a uniform distribution on part orientation. We used a mesh model of a Zymark parallel-jaw gripper with custom fingers as the gripper $\mathcal{R}$, and performed collision checking in OpenRAVE [8]. For computing grasp poses we set θ such that the approach axis $\mathbf{w}$ was maximally aligned with the table normal given the stable pose. Evaluation of P_F was performed with 25 random samples using the Monte-Carlo integration method [15].

A. Label Selection

We used human labels to mask features (holes, air flows, or connectors) of each part to reflect the coverage metrics and tradeoffs that might be observed in practice, since proprietary features are often masked by hand in industry. A single human user without prior knowledge of the details of the Privacy-Preserving Grasp Planning algorithm used our tool to label each of the 23 parts with a single proprietary zone and also labeled four of the parts with a set of five disjoint masks to study the privacy-coverage tradeoff. The user was instructed to label the largest feature on the part surface of each as proprietary for the single masks, and to mask the five largest features in arbitrary order for the nested masks.

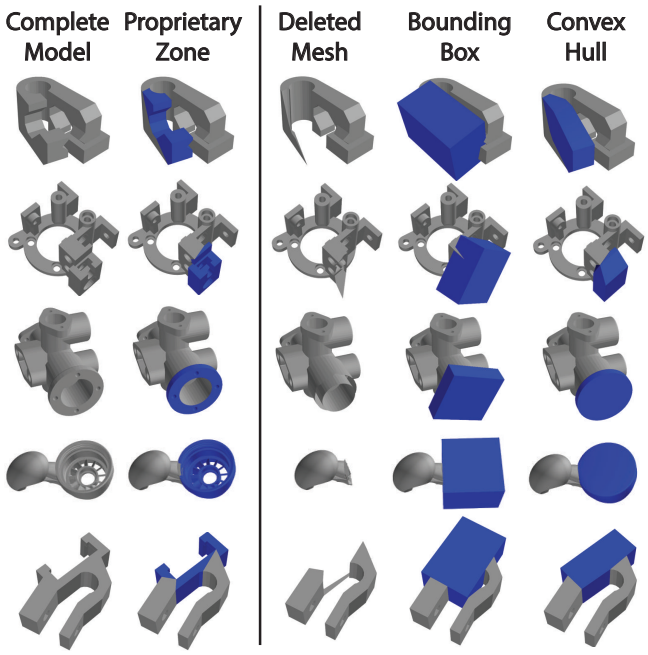

Fig. 4: Illustration of the different methods for masking proprietary zones on five example parts: (top to bottom) a bar clamp, a gearbox, a pipe connector, a turbine housing, and an endstop holder. (Left to right) The original part geometry and the geometry with the proprietary zone highlighted. One method to mask parts is to delete the proprietary zone of the mesh, however this can lead to planned grasps in collision on the original object. Alternatively, the proprietary zone can be replaced with a its bounding box or convex hull.

B. Comparison of Masking Methods

Table I compares each of the masking methods from Section V in terms of the average coverage metric for the single mask dataset over the stable poses of the 23 parts, the percentage of planned grasps that are in collision on the true object, and the Multi-View Convolutional Neural Network object kernel similarity metric from Dex-Net 1.0 [26]. High similarity to the original object indicates that the masked mesh could be used to accelerate grasp planning for new objects with prior data. We see the method of deleting the proprietary region of the mesh performs well in terms of coverage but leads to planned grasps in collision on the original object, which could be problematic if the grasps were executed without further checks. Fig. 5 illustrates some of these failure modes. Grasps planned on the convex hull masked parts are never in collision on the original part and provide higher coverage and higher similarity to the original object

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

Masking Method	Mean α	% Collision	Similarity
Mesh Deletion	**0.79**	6.9	1.05
Bounding Box	0.70	**0.0**	2.60
Convex Hull	0.74	**0.0**	**3.22**

TABLE I: Evaluation on the 23 test parts for masking by removing vertices, replacing the proprietary region with a bounding box, and replacing the proprietary region with a convex hull. The mean coverage α over all objects is best for mesh deletion, however the planner may return grasps that are in collision on the nominal part. Since both the bounding box and convex hull are supersets of the original geometry, neither leads to any grasps in collision. Of the two, the convex hull method performs better in average coverage and similarity to the original object according to the MV-CNN similarity metric of Dex-Net 1.0.

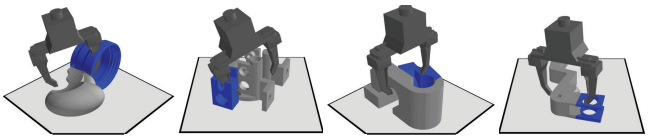

Fig. 5: Illustration of grasps in collision planned on a part using the mesh deletion method. Failures occur because the true geometry of the private part blocks access of a gripper to the planned contacts.

than the bounding box method, suggesting that speedups with prior data observed in Dex-Net 1.0 [26] would hold.

C. Privacy-Coverage and Robustness-Coverage Tradeoff

Fig. 6 studies the privacy-coverage tradeoff and robustness coverage tradeoff on a set of four parts (a gearbox, an extruder, a nozzle mount, and an idler mount), each with five disjoint proprietary regions masked using the convex hull method.

For the privacy-coverage tradeoff we compared $\alpha(\Gamma, \mathcal{E}(\mathcal{R}, S_i, \tau))$ for the stable pose with the highest probability and $\tau = 0.01$ to the privacy metric γ. We see that coverage never increases with increased privacy, consistent with the theory of Section IV. However, the rate of change of coverage with respect to privacy does not appear to be consistent across the examples. This may be because grasps do not appear to be uniformly distributed across the part surface, suggesting that removing some parts of the mesh can affect coverage more significantly than others. This effect is illustrated in the covering sets displayed in Fig. 7.

For the robustness-coverage tradeoff we ran the privacy-preserving grasp planning algorithm with a fixed privacy mask and robustness values $\tau \in [0, 1]$ in increments of 0.05. We compared $\alpha(\Gamma, \mathcal{E}(\mathcal{R}, S_i, 0))$ for the stable pose with the highest probability to the robustness τ for Γ planned by the algorithm. We see that the coverage always decreases with an increasing robustness threshold, consistent with the intuition that the set of possible grasps considered by our algorithm can only decrease with increasing τ.

D. Covering Grasp Sets

Fig. 7 compares the top 50 most robust grasps from the covering grasp sets for the original masked part versus the grasp set computed by our algorithm using convex hull masking for a set of eight example parts. We see that for several parts, such as the fan shroud and turbine housing, the set of most robust

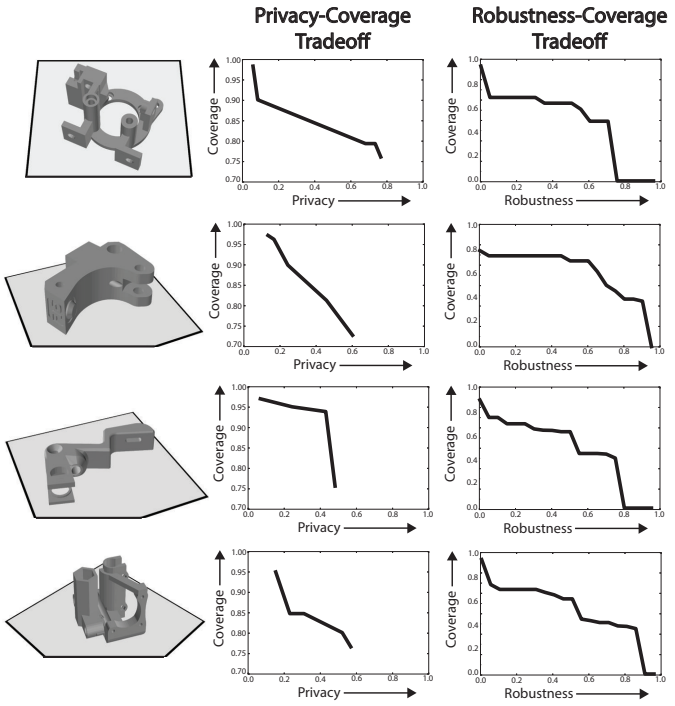

Fig. 6: Study of the privacy-coverage tradeoff and robustness-coverage tradeoff for four example parts (a gearbox, an extruder, a nozzle mount, and an idler mount), each with a sequence of nested proprietary regions. For each part the coverage α never decreases with increasing privacy γ. However, the rate of change of coverage with respect to privacy is not constant, even within a single part. This may be because the set of executable grasps may be more dense in particular regions of the mesh, and jumps occur when areas of high density are masked. Also, coverage always decreases with an increasing robustness threshold.

grasps is clustered in particular regions of the part geometry and when this zone is not masked, the coverage remains high. The covering grasp sets on the original part geometry exhibit variations in density, which may explain the part-variation in the privacy-coverage tradeoffs reported in Section VI-C. Our algorithm correctly avoids the proprietary region of the part and prunes grasps in collision near the table and ares of complex part geometry.

E. Computation Times

The runtimes in minutes for the Privacy-Preserving Grasp Planning algorithm on the eight parts in Fig. 7 were (left to right, top to bottom): 40.0, 36.5, 38.6, 39.1, 41.2, 42.0, 39.6, and 48.9. One average planning took 0.25 seconds per grasp, consistent with the brute force evaluation results reported in [26]. All planning was performed on an Intel Core i7-4770K 3.5 GHz processor with 6 cores.

VII. Discussion and Future Work

We presented an algorithm and system for privacy-preserving grasp planning to find a set of robust grasps for parts while preserving proprietary geometric features of parts such as mounts, connectors, and holes for air flows. Our algorithm masks the part using the convex hull of the proprietary region and evaluates contact pairs on triangles from the

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 19, 2016.

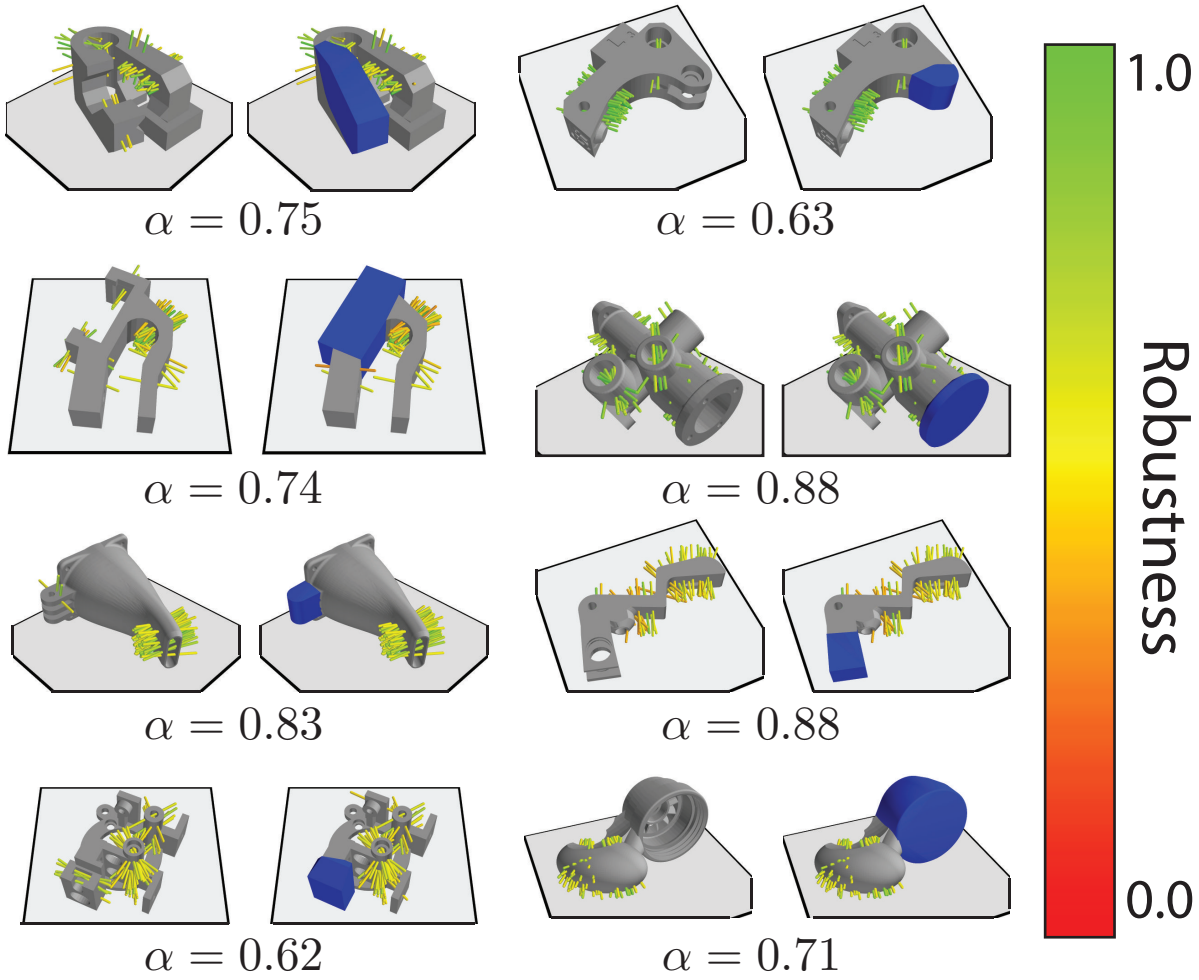

Fig. 7: Comparison of the top 50 most robust grasps from the executable grasp set on the original part and from our algorithm using convex hull masking on eight parts. The coverage α is reported for each of our computed privacy-preserving grasp sets, and proprietary zones are marked in blue. Each grasp axis is colored by its robustness, or probability of force closure (P_F) under uncertainty in object pose, gripper pose, and friction. We see that grasp sets planned by our algorithm are similar to those planned without considering privacy, and that the computed sets do not intersect the private region of the original mesh, suggesting that grasps planned on the masked part preserve privacy and cover the public region of the original part, and that robust grasps tend to be clustered together on certain areas of the part.

public region of the part surface, checking collisions and the probability of force closure for each. We also introduce grasp coverage based on dispersion and prove that coverage cannot increase with increasing part privacy for a sufficient number of grasps input to our algorithm. Experiments suggest that the convex hull masking method outperforms mesh deletion and bounding box masking and that coverage decreases with increasing privacy, and the increase appears to be proportional to the density of grasps in the private region of the mesh.

In future work we will further study the privacy-coverage tradeoff with additional parts and work with industrial experts to refine the privacy-labeling interface and perform physical experiments. We will investigate approaches to increasing computational efficiency by actively identifying candidate grasp surfaces that lack coverage, for example using annealing [6] or Multi-Armed Bandits [22]. We will also explore alternate methods to preserve privacy other than masking, for example adding small deformations to the geometry using decimation or smoothing [37] and investigate how these affect grasp robustness.

VIII. Acknowledgments

This research was performed in UC Berkeley's Automation Sciences Lab under the UC Berkeley Center for Information Technology in the Interest of Society (CITRIS) "People and Robots" Initiative: http://robotics.citris-uc.org. The authors were supported in part by the U.S. National Science Foundation under NRI Award IIS-1227536, by grants from Siemens, UC Berkeley's Algorithms, Machines, and People Lab; the Knut and Alice Wallenberg Foundation; the NSF-Graduate Research Fellowship; and the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. We thank our colleagues who gave feedback and suggestions, in particular Stefano Carpin, Mike Franklin, Animesh Garg, Sanjay Krishnan, Michael Laskey, Reza Moazzezi, Zoe McCarthy, Lauren Miller, Daniel Seita, and Nan Tian.

REFERENCES

[1] "Autodesk inventor," https://forums.autodesk.com/t5/inventor-general-discussion/defeaturing-function-in-inventor/td-p/5433898, 2016.

[2] "Maketbot thingiverse," http://www.thingiverse.com/, 2016.

[3] "Solidworks," http://www.solidworks.com/sw/products/3d-cad/defeature.htm, 2016.

[4] C. C. Aggarwal and S. Y. Philip, *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[5] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2851–2858.

[6] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[7] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater, "Learning grasp affordance densities," *Paladyn, Journal of Behavioral Robotics*, vol. 2, no. 1, pp. 1–17, 2011.

[8] R. Diankov and J. Kuffner, "Openrave: A planning architecture for autonomous robotics," *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-08-34*, vol. 79, 2008.

[9] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," 2012.

[10] S. Gao, W. Zhao, H. Lin, F. Yang, and X. Chen, "Feature suppression based cad mesh model simplification," *Computer-Aided Design*, vol. 42, no. 12, pp. 1178–1188, 2010.

[11] C. Goldfeder and P. K. Allen, "Data-driven grasping," *Autonomous Robots*, vol. 31, no. 1, pp. 1–20, 2011.

[12] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 1710–1716.

[13] M. J. Hutchins, R. Bhinge, M. K. Micali, S. L. Robinson, J. W. Sutherland, and D. Dornfeld, "Framework for identifying cybersecurity risks in manufacturing," *Procedia Manufacturing*, vol. 1, pp. 47–63, 2015.

[14] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015.

[15] B. Kehoe, D. Berenson, and K. Goldberg, "Estimating part tolerance bounds based on adaptive cloud-based grasp planning with slip," in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*. IEEE, 2012, pp. 1106–1113.

[16] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4263–4270.

[17] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 2, pp. 398–409, 2015.

[18] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[19] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security & Privacy*, no. 1, pp. 81–85, 2010.

[20] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni, and R. Scopigno, "Protected interactive 3d graphics via remote rendering," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 695–703.

[21] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3993–3998.

[22] M. Laskey, J. Mahler, Z. McCarthy, F. Pokorny, S. Patil, J. van den Berg, D. Kragic, P. Abbeel, and K. Goldberg, "Multi-arm bandit models for 2d sample based grasp planning with uncertainty." in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*. IEEE, 2015.

[23] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[24] S. R. Lindemann and S. M. LaValle, "Incrementally reducing dispersion by increasing voronoi bias in rrts," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3251–3257.

[25] J. Mahler, S. Patil, B. Kehoe, J. van den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning under shape uncertainty using gaussian process implicit surfaces and sequential convex programming," 2015.

[26] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kroeger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," 2016.

[27] S. Marras, F. Ganovelli, P. Cignoni, R. Scateni, and R. Scopigno, "Controlled and adaptive mesh zippering." in *GRAPP*, 2010, pp. 104–109.

[28] R. Mason, E. Rimon, and J. Burdick, "Stable poses of 3-dimensional objects," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 391–398.

[29] A. V. Mobley, M. P. Carroll, and S. A. Canann, "An object oriented approach to geometry defeaturing for finite element meshing." in *IMR*, 1998, pp. 547–563.

[30] H. Niederreiter, "Quasi-monte carlo methods and pseudo-random numbers," *Bulletin of the American Mathematical Society*, vol. 84, no. 6, pp. 957–1041, 1978.

[31] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama, "Watermarking 3d polygonal meshes in the mesh spectral domain," in *Graphics interface*, vol. 2001. Citeseer, 2001, pp. 9–17.

[32] P. Oswald and P. Schröder, "Composite primal/dual 3-subdivision schemes," *Computer Aided Geometric Design*, vol. 20, no. 3, pp. 135–164, 2003.

[33] F. Panahi, M. Davoodi, and A. F. van der Stappen, "Orienting parts with shape variation," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 479–496.

[34] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," *arXiv preprint arXiv:1509.06825*, 2015.

[35] F. T. Pokorny and D. Kragic, "Classical grasp quality evaluation: New theory and algorithms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[36] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, no. 1, pp. 69–73, 2012.

[37] A. Thakur, A. G. Banerjee, and S. K. Gupta, "A survey of cad model simplification techniques for physics-based simulation applications," *Computer-Aided Design*, vol. 41, no. 2, pp. 65–80, 2009.

[38] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 1994, pp. 311–318.

[39] M. Vahedi and A. F. van der Stappen, "Caging polygons with two and three fingers," *Int. J. Robotics Research (IJRR)*, vol. 27, no. 11-12, pp. 1308–1324, 2008.

[40] T. P. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2106–2113.

[41] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2012, pp. 557–562.

[42] J. Wiegley, A. Rao, and K. Goldberg, "Computing a statistical distribution of stable poses for a polyhedron," in *PROCEEDINGS OF THE ANNUAL ALLERTON CONFERENCE ON COMMUNICATION CONTROL AND COMPUTING*, vol. 30. UNIVERSITY OF ILLINOIS, 1992, pp. 836–836.

[43] A. Yershova and S. M. LaValle, "Deterministic sampling methods for spheres and so (3)," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3974–3980.

[44] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005.