

# Planning Grasps With Suction Cups and Parallel Grippers Using Superimposed Segmentation of Object Meshes

Weiwei Wan , *Member, IEEE*, Kensuke Harada , *Member, IEEE*, and Fumio Kanehiro , *Member, IEEE*

**Abstract**—This article develops model-based grasp planning algorithms. It focuses on industrial end-effectors like grippers and suction cups, and plans grasp configurations considering computer aided design (CAD) models of target objects. The developed algorithms can stably find many high-quality grasps, with satisfying precision and little dependency on the quality of CAD models. The undergoing core technique is superimposed segmentation, which preprocesses a mesh model by peeling it into superimposed facets. The algorithms use the facets to locate contacts and synthesize grasp poses for popular industrial end-effectors. Several tunable parameters are prepared to adapt the algorithms to meet various requirements. The experimental section studies the influence of the tunable parameters and analyzes the cost, precision, and robustness of the proposed algorithms and their planned grasps, with both simulations and real-world systems. Besides, the proposed algorithms are applicable to mesh models reconstructed from point clouds obtained by depth sensors. Some experiments and analysis are also carried out to study and demonstrate the ability.

**Index Terms**—Grasping, grippers and other end-effectors, manipulation planning.

## I. INTRODUCTION

**T**HIS article develops algorithms to plan grasping configurations automatically. It focuses on industrial end-effectors, and plans grasp poses for these end-effectors considering computer aided design (CAD) models of target objects.

Developing grasp planning algorithms is vital to manufacturing using “teachingless” robotic manipulators. Modern robotic manipulation systems use manually annotated or pretaught grasp configurations to perform specific tasks, which is not only costly

but also difficult to redeploy. Automatic grasp synthesis or planning algorithms could bypass the annoying manual work and enable fast redeployment for varying and changing manufacture. For this reason, lots of studies in the field of robotic grasping have been devoted to automatic grasp planning, and many grasp planning algorithms have been developed. These algorithms can plan grasps considering forces and collisions. However, they hardly meet the requirements of fully automatic manufacturing applications like bin-picking and assembly. The requirements include but are not limited to: 1) flexibility: the grasp planner is expected to provide as many candidate grasp poses as possible for optimization; 2) robustness: the grasp planner must have little dependency on the quality of CAD models; and 3) precision: the relative object poses do not change much after being grasped by the planned grasps.

On the other hand, on-going grasp planning studies concentrate on theoretical aspects like grasp closures and qualities, or applicational aspects like grasping using multifinger hands, dexterous hands, and hands with tactile and force sensors. Grasp planning for popular industrial end-effectors, e.g., parallel grippers and suction cups, is usually ignored since grasping rigid objects using such end-effectors is considered to be easy and solved. In this article, we reinspect this opinion and restudy the grasp planning problem for parallel grippers. After reviewing previous work, we found that although grasp planner for parallel grippers had existed for decades, they do not really meet the requirements of industrial tasks: some old-fashioned algorithms could plan grasps for simple polytopes (e.g., ray-shooting-based methods, simple segmentation-based methods, etc.), but they do not apply to complicated mesh models and cannot output a satisfying number of grasps; and modern grasp planning algorithms aim to find stable grasps, but they are short in finding a large amount of grasping poses and tend to overlook the precision or the robustness of the planned grasps under different mesh qualities. It is thus difficult to use the planned grasps for industrial tasks like bin-picking and assembly.

Under this background, this article develops new grasp planning algorithms for industrial bin picking and assembly. The problem is formulated as follows. The goal is to develop algorithms that accept: 1) kinematic models of industrial end-effectors like suction cups, parallel grippers, and three-finger-one-parameter grippers; and 2) CAD models of rigid objects. The algorithms are expected to produce: A set of automatically planned grasp configurations. We assume: 1) the objects have

Manuscript received February 24, 2020; revised June 3, 2020; accepted July 19, 2020. Date of publication August 18, 2020; date of current version February 4, 2021. This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This article was recommended for publication by Associate Editor L. Natale and Editor M. Yim upon evaluation of the reviewers' comments. (*Corresponding author: Weiwei Wan.*)

Weiwei Wan and Kensuke Harada are with the Graduate School of Engineering Science, Osaka University, Osaka 565-0871, Japan (e-mail: wan@sys.es.osaka-u.ac.jp; harada@sys.es.osaka-u.ac.jp).

Fumio Kanehiro is with the National Institute of Advanced Industrial Science and Technology AIST, Ibaraki 305-8560, Japan (e-mail: f-kanehiro@aist.go.jp).

This article has supplementary downloadable material available at <https://ieeexplore.ieee.org>, provided by the authors. The material consists of a video with a special focus on the grasp planning process.

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2020.3014036

rigid bodies. Soft or changeable objects are not considered; and 2) the end-effectors and manipulators are actuated using position control to ensure fast operation. Tactile or *F/T* sensors are not available.

The fundamental technique of the developed algorithms is superimposed segmentation, which is specially designed to preprocess the mesh models for contact and grasp planning. The superimposed segmentation preprocesses a mesh model by peeling it into facets. Each facet is allowed to overlap with others, and thus the peeling preprocessing is called superimposed segmentation. The overlap and sizes of the peeled facets are controlled by several tunable parameters, which allow users to change the quality of planned grasps following the requirements of their applications. The superimposed facets are used to determine contacts. Grasp poses for popular industrial end-effectors are planned considering the contacts.

The manuscript presents the details of the superimposed segmentation, the tunable parameters, as well as the planning algorithms. It carries out experiments to compare and analyze the roles of the parameters and the performances of the proposed planner. Also, some real-world regrasp and assembly systems are implemented to show the efficacy of the proposed algorithms. The experimental results and real-world integrations show that the developed algorithms could plan a large number of precise grasps with little dependency on CAD models' quality (triangle densities). Object positions change less than 2 mm after being grasped the planned grasps. Besides, the algorithms are applicable to mesh models reconstructed from point clouds obtained by depth sensors. They meet the requirements for modern industry bin-picking and assembly. Compared to popular methods using ray-shooting or simple segmentation, the algorithm is more flexible, robust, and precise.

The rest of this article is organized as follows. Section II reviews related work. Section III discusses the fundamental technique like superimposed segmentation and sampling contact points. In Section IV, details of grasp planning algorithms using the fundamental technique are shown. Section V presents the experiments and analysis. Finally, Section VI concludes this article.

## II. RELATED WORK

This article develops model-based grasp planning algorithms for suction cups and parallel grippers. Accordingly, this section reviews related studies on grasp theories and grasp planning of suction cups and parallel grippers, with a particular focus on preprocessing mesh models. Also, since the proposed algorithms form a basic component of our open-source robotic simulator, it is used by many projects and appeared in several previous publications. The difference with them is clarified in the end.

### A. Grasp Theories and Grasp Planning

Grasp theories study form/force closure and closure qualities. The theoretical studies apply to suction cups, parallel grippers, as well as other robotic hands. Some early work like [1] and [2] studied point fingers and polygonal objects, with later extensions to more realistic scenarios like curved

surfaces and fingers [3]–[7], considering grasp stability [8]–[10], and grasp metrics [11], [12]. The early theoretical studies were mostly 2-D, and the concentration was to estimate the stability of grasps and the resistance to external wrenches. The theoretical studies were extended to 3-D polyhedral objects or mesh models composed of flat faces later, assuming to be hard point contacts. Examples include [13], [14], etc. Some other studies optimized the planned grasps [15] using some quality metrics [16].

Planning grasp poses for real-world objects and real-world end-effectors are more challenging than the early theoretical studies. There is a big gap between the computed results and real-world executions. Many factors like contact regions, object surface curvatures, resistance to torque caused by gravity forces, kinematics of robot hands, etc., must be considered to secure stable and precise grasps. Several previous studies challenged these difficulties. For example, Wolter *et al.* [17] considered the geometry of grippers during the automatic generation of grasps for 3-D rectilinear objects. Jones *et al.* [18] considered the parallel faces of a 3-D object as well as the mesh model of a robot gripper to plan two-finger grasps for pick-and-place operations. Liu *et al.* [19] used the attractive regions of an object to plan stable grasps. Pozzi *et al.* [20] discussed grasp qualities considering the kinematic structures of underactuated and compliant hands. Shi *et al.* [21] considered environmental constraints as well as the kinematic constraints of robot hands to plan accessible grasps for bin-picking and kitting tasks. Li *et al.* [22] used stretching ropes (cord geometry) to find the contact of a hand jaw with object surfaces and hence plan the grasps. Ciocarlie *et al.* [23] considered local geometry and structures at contact points and modeled friction forces using soft models. Harada *et al.* [24] discussed a gripper with soft finger pads attached to the fingertips and analyzed object mesh models considering the depth of contacts. These studies used gripper models and their kinematic structures to ensure feasibility and considered contact properties by analyzing the meshes around contact regions.

This article plans grasp for suction cups and parallel grippers. For these simple end-effectors, the form/force closure theory is simplified to comparing surface normals at the contacts. The kinematic constraints, contact, and quality of grasps are considered in segmentation, sampling, and nested collision detection, respectively. The planned grasps quality depends on the preprocessing of mesh models, which is further reviewed below. With these adaptations, the algorithms proposed in this article could plan a large number of precise candidate grasps with little influence by CAD mesh qualities. They are ready to be used by an integrated task and manipulation planner for high-level planning.

### B. Preprocessing Mesh Models

Two major approaches to preprocess the mesh models for grasp planning are 1) primitive fitting and 2) surface segmentation. The first approach represents mesh models using a set of shape primitives, and plans grasp by considering the fitting errors or using preannotated grasps. The second approach represents mesh models using coplanar triangle sets. Each coplanar

segmented triangle set is named a facet and equals to one constitutional polygon of a polyhedron.

For primitive fitting-based grasp planning, Goldfeder *et al.* [25] represented a mesh model using recursive splitting and fitting of primitive superquadrics [26]. El-Khoury *et al.* [27] fitted segmented point clouds using primitive superquadrics and used preannotated training sets to learn grasp points from the fitted models. Xue *et al.* [28] also used primitive superquadrics to fit models and plan grasps for Schunk Anthropomorphic Hands. Other than superquadrics-based fitting, Miller *et al.* [29] represented a mesh model using a set of primitive mesh models like boxes and spheres, and use a set of rules defined on the primitives to generate grasps for the mesh model. Hueber *et al.* [30] fitted mesh models using different levels of primitive boxes and planned grasp by evaluating the annotated grasps on the primitives. Bonilla *et al.* [31] also fitted mesh models using primitive boxes and planned grasps using geometric information extracted from the primitive boxes. Nagata *et al.* [32] proposed an interactive method for grasp planning by assuming shape primitives. Yamanobe *et al.* [33] defined the gripping configurations of several shape primitives and used primitive shape representation to plan grasping poses for mobile manipulators. Curtis *et al.* [34] used primitives to learn grasps. Instead of explicitly fitting primitives, the authors used learned grasping knowledge on a set of primitive objects to speed up the process of planning successful grasps for novel objects. Harada *et al.* [35] used cylinders to fit banana point clouds, and planned robust grasps by analyzing the projections of the point clouds on the cylindrical axes denoted by the fitted cylinders. The grasp moduli space proposed by Pokorný *et al.* [36] is also a primitive fitting approach. The primitive fitting approaches do not assure the stability of planned grasps, or the exact object poses after grasping. Some further evaluations or optimizations are needed to make the results practical.

For surface segmentation-based grasp planning, Harada *et al.* [24] clustered triangle meshes by using a parameter denoting softness of contacts and implemented grasp planning for grippers with soft finger pads. Tsuji *et al.* [37] used multilevel clustering [38] to find the concavity and convexity of mesh models, and used stress distribution models to plan stable grasps. Hang *et al.* [39] also used multilevel clustering to plan grasps. The difference is their goal was not to find grasping features. Instead, they use different levels of simplification to iteratively search for stable grasps under reachability constraints. In a later work, Hang *et al.* [40] extended the study to fingertip spaces and used multiresolution contacts to expedite grasp synthesis. The results of the multilevel planning were demonstrated in [41] using an Allegro hand. The hand could gait to different configurations as the weights of objects change. Some of the primitive fitting approaches also have a segmentation step, where meshes or point clouds are segmented for fitting [27], [42].

The algorithms developed in this article use surface segmentation to plan contacts and estimate closure qualities. Unlike previous work that segmented each triangle into a single facet or performed multilevel segmentation, we propose the idea of superimposed segmentation. Each triangle is repeatedly segmented into different facets, and the overlap and thickness of the

facets are controlled by tunable parameters pertaining to surface normals. The advantage of the superimposition is it produces many uniform facets, allows sampling more contacts, and thus enables planning a large number of stable and precise grasp configurations for suction cups and parallel grippers. The grasp planning algorithms using the superimposed segmentation decouple the triangle density and levels of details from the grippers and suction cups, making the planned results less influenced by different mesh qualities.

### C. Relations to Our Previous Studies

Previously, many projects were developed using grasping poses planned by the algorithms in this article. These projects use the planned grasps to iteratively select start and goal poses in the robot workspace [43]–[45], build manipulation graphs [46]–[50], or optimize object assembly sequences [51]–[53]. Although the proposed algorithms were heavily used in these projects, they were not carefully introduced or analyzed. The related publications focused on high-level task and motion planning problems instead of the fundamental grasp planning. This article fills up the gap. It presents all details of the grasp planner, discusses the tunable parameters, their precision, and robustness under various mesh qualities, as well as the applications to mesh models reconstructed from sensor-collected point clouds. It also shows some of our projects that use the planned grasps to pick up and manipulate objects at the end of this article. The projects are compiled into a supplementary video with a special focus on the grasp planning process.

## III. PREPROCESSING MESH MODELS

### A. Superimposed Segmentation

Superimposed segmentation provides uniform facets. Here, by uniform, we mean facets peeled from a surface with similar geometry features are equally large. Conventional approaches [38], [54] cluster each triangle<sup>1</sup> into a single facet, resulting in uneven facets—Some of them could be large, while others are small, even if the facets are all from a surface with similar geometry feature. Unlike the conventional approaches, superimposed segmentation allows one triangle to be repeatedly clustered to multiple facets, making all facets uniform. The clustered facets do not exclusively occupy the triangles that might also belong to other facets. Fig. 1 shows an example of this idea.

The superimposed segmentation is computed as follows. First, the algorithm randomly initiates a seed triangle and scans the surrounding triangles of the seed. See Fig. 2(a) for example. The purple triangle is the seed triangle, and the algorithm scans the triangles surrounding it. If the angle between the normals of the seed triangle and an adjacent triangle is smaller than a threshold  $\theta_{\text{pln}}$ , the adjacent triangle is clustered into the same facet as the seed triangle. In the figure, the angles between the black arrows and the purple arrow are equal to or smaller than  $\theta_{\text{pln}}$ . Thus, the related triangles are clustered into the purple facet.

<sup>1</sup>We assume the CAD is triangulated. If not, the surface triangulation methods introduced in [55] could be used to triangulate mesh models.



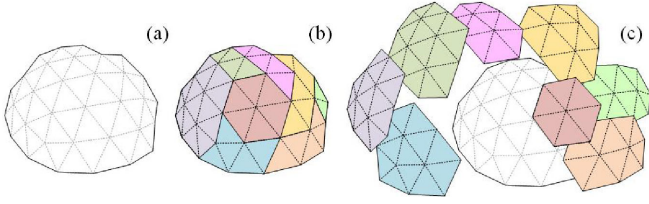


Fig. 1. Segmenting mesh models into superimposed facets. (a) Original mesh model. (b) Results of segmentation. (c) The facets are superimposed.

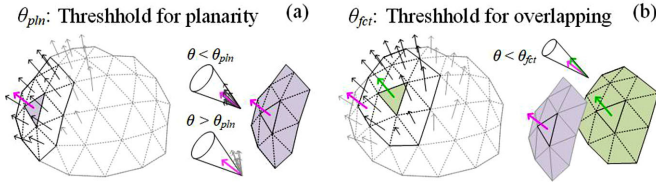


Fig. 2. Superimposition is controlled by two parameters. (a)  $\theta_{pln}$  controls planarity of each facet. (b)  $\theta_{fct}$  controls the overlap of facets.

In contrast, the angles between the grey arrows and the purple arrow are larger than  $\theta_{pln}$ . Consequently, the related triangles are not included.  $\theta_{pln}$  is a tunable parameter that controls the planarity of a facet.

After clustering the first facet, the algorithm randomizes a new seed triangle and repeats the clustering by starting from the new seed. The routine to randomize a new seed is as follows. The algorithm scans the surrounding triangles of the previous seeds and checks the angles between the previous seeds' normals and the normals of the surrounding triangles. If an angle is larger than  $\theta_{fct}$ , the related triangle is selected as the new seed. Fig. 2(b) shows an example the process. The angle between the green normal and the purple normal is larger than  $\theta_{fct}$ . Thus, the triangle with the green normal is selected as the new seed. The algorithm repeats the clustering process by using a new seed and generates a new facet [the green facet shown in the right part of Fig. 2(b)].  $\theta_{fct}$  is a tunable parameter that controls the superimposition of facets.

During clustering, all triangles are repeatedly scanned, which allows one triangle to be clustered into multiple facets. The facets could superimpose with each other. On the other hand, the normals of all previous seeds are compared when initiating a new seed, which ensures each facet is unique, and the algorithm can stop properly.

## B. Sampling Contact Points

1) *Sampling and Distributing*: Contact points are computed by sampling the surface of the object mesh model. The sampling is first performed over the whole surface to provide evenly distributed contact points on the mesh. Then, the sampled points are repeatedly distributed to the superimposed facets as their contact points. Note that we avoided sampling individual facets, since it only provides evenly distributed contact points on individual facets, the overall distribution relies on the segmentation

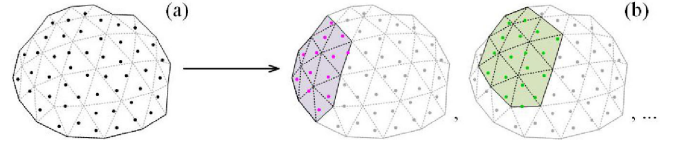


Fig. 3. (a) Sample over the whole mesh surface. (b) Distribute the samples to each facet to avoid repeated sampling.

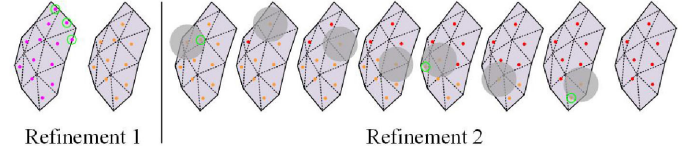


Fig. 4. Two refinements that remove the bad contact points. The first refinement removes the contact points with small distances to facet boundaries. The second refinement removes the contact points that are too close to others.

methods. In cases, where a facet is small, individual sampling may fail to produce contact points.

Fig. 3 shows an example. In Fig. 3(a), the whole mesh surface is sampled and in Fig. 3(b), the sampled points are distributed to superimposed facets. The surface is sampled once, and the sampled points are distributed to individual facets repeatedly. The method ensures the contact points on each facet have equal density and are evenly distributed. It is irrelevant to the segmentation methods. Also, the method distributes the samples to multiple facets without sampling again. It avoids repeated computation and improves algorithm efficiency.

2) *Removing Bad Samples*: The output of sampling and distributing cannot be used directly, since the distributed samples might be 1) near the boundary of facets and 2) near to each other. In the first case, attaching fingerpads to the sampled points near facet boundaries lead to unstable grasps. In the second case, attaching fingerpads to the near points produces similar grasping configurations, which results in a large number of similar grasps and is wasteful. To avoid the problems, we perform two refining processes, where the first one computes the distance between a contact point and the boundary of its facet. The points with distances smaller than  $t_{bdry}$  are removed. The second one removes the contact points that are too close to others. The remaining contact points after being refined by the first process are further screened using the fixed-radius nearest neighbor (frNN) algorithm to remove nearby points with a distance smaller than  $t_{rnn}$ . Like  $\theta_{pln}$  and  $\theta_{fct}$ ,  $t_{bdry}$  and  $t_{rnn}$  are tunable parameters of the grasp planner. Note that both the two refining processes are done locally in a facet. The overlapped samples on superimposed facets are not affected. Fig. 4 shows a graphical view of the two refining processes.

In practice,  $t_{rnn}$  is determined by the size of finger pads. An end-effector contacts with objects at a region, instead of a single point.  $t_{rnn}$  specifies the radius of the contact region. It controls the density of planned grasps by removing nearby candidates.

As a demonstration, Fig. 5 shows the process of sampling contact points using a plastic workpiece shown in Fig. 5(a).



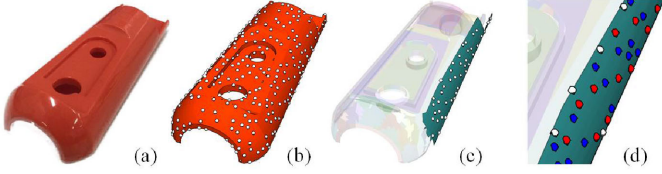


Fig. 5. Sampling contact points. (a) The original object. (b) The sampled contact points. (c) The contact points distributed to one facet. (d) Removing bad contact points. Especially in (d), the white points are removed since they are too near to the boundary. The red points are the results of frNN screening.

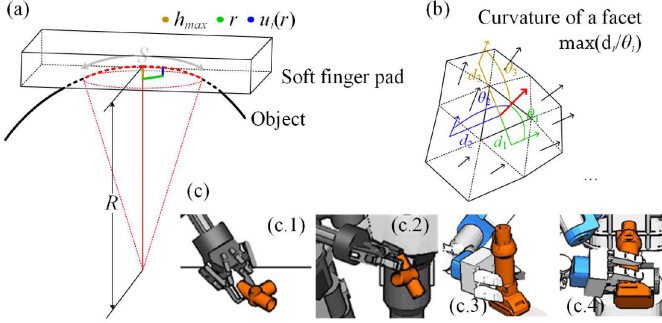


Fig. 6. (a) SFC model. (b) The curvature of a facet. (c) The goal is to make sure the object is stable at an arbitrary pose.

3) *Stability*: The soft-finger contact (SFC) model proposed in [56] is used to estimate the stability of a grasp. The force and torque exerted by one SFC are expressed as follows:

$$f_t^T f_t + \frac{\tau_n^2}{e_n^2} \leq \mu^2 f_n^2. \quad (1)$$

Here,  $f_t$  indicates the tangential force at the contact.  $\tau_n^2$  indicates the torque at the contact.  $f_n$  indicates the load applied in the direction of the contact normal.  $e_n$  is the eccentricity parameter that captures the relationship between maximum frictional force and moment. Under the Winkler elastic foundation model,  $e_n$  is

$$e_n = \frac{\max(\tau_n)}{\max(f_t)} = \frac{\int_S r \mu K u_i(r) dS}{\int_S \mu K u_i(r) dS} \quad (2)$$

where  $K$  is the elastic modulus of the foundation over the thickness of the soft finger pad,  $S$  is the contact surface between the finger pad and the object,  $r$  is the distance between a differential contact point and the center of the contact region, and  $u_i(r)$  is the depth of the soft penetration. These symbols are illustrated in Fig. 6(a). The area marked by the dashed red curve shows the contact surface. The area spanned by the gray arrows shows the contact surface  $S$ . The green segment shows the distance  $r$ . The blue segment shows the penetration depth  $u_i(r)$ . Using  $R$  to denote the radius of the contact curvature,  $u_i(r)$  can be represented as follows:

$$u_i(r) = \sqrt{R^2 - r^2} - (R - h_{\max}) \quad (3)$$

where  $h_{\max}$  is the maximum depth of the soft penetration. The golden segment in Fig. 6(a) illustrates it.  $R$  could be computed using  $\max(d_i/\theta_i)$  following the definition of curvature, where  $d_i$  is the distance between the center of the  $i$ th facet and the

center of the seed triangle,  $\theta_i$  is the angle between the normal of the  $i$ th triangle and the normal of the seed triangle. The concrete meaning of  $d_i$  and  $\theta_i$  on a facet is illustrated in Fig. 6(b). Here, three triangles identified by green, golden, and blue normals are used for illustration. The  $d_i$ s of these triangles are essentially the lengths of their connecting segments to the center of the seed. The  $\theta_i$  of the triangles are essentially the in-between angles between the normals. The  $R$  of a facet is thus computed as the maximum  $d_i/\theta_i$  of all triangles in the facet.

$h_{\max}$  is used as a tunable parameter to control the stability of planned grasps. Since the goal of stability estimation is to make sure the object is stable at an arbitrary pose [see Fig. 6(c)], a planned grasp must meet<sup>2</sup>

$$(mgc)^2 \leq e_n^2 (\mu^2 f_n^2 - (mg)^2) \quad (4)$$

where  $c$  is the distance between the center of mass (com) of the object and the center of contacts. During manipulation, the largest external torque appears when gravity direction is perpendicular to vector  $\overrightarrow{\text{contact} - \text{com}}$ . From (2), (3), and (4), we obtain

$$(mgc)^2 \leq \left(\frac{8}{15}\right)^2 (2Rh_{\max} - h_{\max}^2) (\mu^2 f_n^2 - (mg)^2) \quad (5)$$

This equation is used to determine the stability of planned grasps. A given  $h_{\max}$  will be substituted into it to check if a planned grasp meets the stability requirements. Only stable ones are kept.

#### IV. PLANNING THE GRASP CONFIGURATIONS

This section develops algorithms to plan grasps for suction cups and parallel grippers using the superimposed facets and the sampled contact points.

##### A. Suction Cups

We assume a suction end-effector has only one suction cup. The grasp planning algorithm for a single suction cup is a two-step process. In the first step, the algorithm finds the possible orientations to attach the suction cup to the sampled and refined contact points. Since the approaching direction must be perpendicular to the contact region, the end-effector's orientation is only changeable by rotating around the approaching direction. The algorithm poses the suction cup to the contact points from the changeable orientations and removes the infeasible (collided) grasps. In the second step, the algorithm examines the resistance of planned suction configurations to external torques caused by gravity. The pseudocode of the algorithm is shown in Algorithm 1.

First, for each contact point, the algorithm discretizes the rotation around the contact normal into *discrete angles*, and computes the rotation matrices. The number of discretized values is determined by  $n_{\text{da}}$ . The algorithm poses the *eemesh* (mesh model of the end effector) using the computed rotation matrices and checks the collision between the *eemesh* and the *objmesh* (mesh model of the object). Line 3 of the pseudocode iterates

<sup>2</sup>Here, we assume a grasp is quasi-static and are considering the worst case, where the gripping torque must resist the largest torque caused by gravity.

**Algorithm 1:** Planning for Suction Cups.

---

```

1 plannedgrasps  $\leftarrow$  []
2 for  $contact, contactNormal \in facet$  do
3   for  $theta \in discreteangles$  do
4     /*size (discreteangles)  $\leftarrow n_{da}$ */
5      $rotmat \leftarrow rodrigues(contactNormal, theta)$ 
6      $eemesh.setPose(contact, rotmat)$ 
7     if not  $checkCollision(eemesh, objmesh)$ 
8       if  $checkTorque(contact, objcom, h_{max})$ 
9          $plannedgrasps.add([contact, rotmat])$ 

```

---

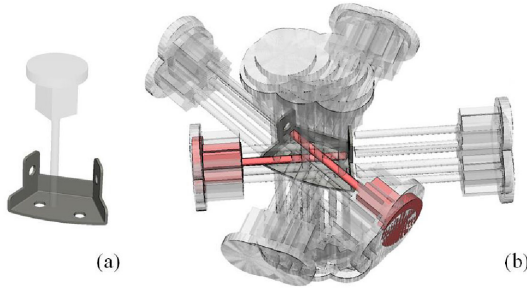


Fig. 7. Example of the grasp planning for a suction cup and a metal workpiece. (a) One of the planned grasps. (b) All results. The red grasps collide with the metal workpiece.

through the discrete orientations. Line 6 poses the *eemesh* to a *contact* with rotation matrix *rotmat*. Line 7 checks the collision between *eemesh* and *objmesh*.

After that, the algorithm examines the resistance to external torques. The function  $checktorque(contact, objcom, h_{max})$  in line 8 of the pseudocode performs the examination. The function computes the Euclidean distance between *contact* and *objcom* (the *com* of the object), and checks if (4) is met.

Fig. 7 shows an example of the planning algorithm using a suction cup and a metal workpiece. Fig. 7(a) is a single planned grasp. Fig. 7(b) are all the planned results. The red configurations in Fig. 7(b) are the grasps deleted by the  $checkcollision(eemesh, objmesh)$  function.

### B. Two-Finger Parallel Grippers

Grasp planning for two-finger parallel grippers is a three-step process. In the first step, the planner for two-finger parallel grippers first finds parallel facets and computes candidate contact pairs by examining the contact points on the parallel facets. Compared with suction cups that need one contact point, the planner for two-finger parallel grippers needs two contact points with opposite contact normals. Thus, the algorithm involves an extra step to prepare candidate contact pairs. The second and third steps are similar to suction cups. In step two, the algorithm finds the possible orientations to attach the parallel gripper to the candidate contact pairs. In step three, the algorithm examines the stability of the planned grasps.

The pseudocode of the grasp planner for two-finger parallel grippers is shown in Algorithm 2. In the first block (see lines

**Algorithm 2:** Planning for Two-Finger Parallel Grippers.

---

```

1 candidatepairs  $\leftarrow$  []
2 for  $(facet_a, facet_b) \in objmesh$  do
3   if  $facetNormal_a \cdot facetNormal_b \leq \cos(\theta_{pari})$ 
4     for  $(contact_a, contactNormal_a) \in facet_a$  do
5        $ray \leftarrow ray(contact_a, -contactNormal_a)$ 
6        $contact_b \leftarrow intersect(ray, facet_b)$ 
7       if  $contact_b$  is not None
8          $candidatepairs.add([contact_a,$ 
9            $contact_b])$ 
9 plannedgrasps  $\leftarrow$  []
10 for  $(contact_a, contact_b) \in candidatepairs$  do
11    $stkmesh.setPose(contact_a, contactNormal_a,$ 
12      $contact_b, contactNormal_b)$ 
13   if not  $checkCollision(stkmesh, objmesh)$ 
14     for  $theta \in discreteangle$  do
15       /*size (discreteangles)  $\leftarrow n_{da}$ */
16        $tipPosition \leftarrow (contact_a + contact_b)/2$ 
17        $rotmat \leftarrow rodrigues(contact_a - contact_b,$ 
18          $theta)$ 
19        $eemesh.setPose(tipPosition, rotmat)$ 
20       if not  $checkCollision(eemesh,$ 
21          $objmesh)$ 
22         if  $checkTorque(tipPosition, objcom,$ 
23            $h_{max})$ 
24            $plannedgrasps.add([contact_a,$ 
25              $contact_b, rotmat])$ 

```

---

1–8), the algorithm finds candidate contact pairs. For each pair of parallel facets, the algorithm initiates a ray that starts from a contact point on one facet and points to the inverse direction of the contact normal. It detects the intersection between the ray and the other facet. If an intersection exists, the contact–intersection pair is saved as a candidate. Whether two facets in a pair are parallel is determined by a tunable parameter  $\theta_{pari}$  (see line 3). In the second block (see lines 10–20), the algorithm performs collision detection and examines stability. The algorithm invokes two nested  $checkcollision()$  functions in the second block. In the first call (see line 12), the algorithm checks if the gripper’s stroke collides with the object. The stroke is represented by cylinders that do not have an orientation. In the second call (see line 18), the algorithm checks if the whole hand (both fingers and palm) collides with the object. Especially, the algorithm poses the gripper by attaching its two fingertips to the contact pair and performs collision detection at different rotations around the axis formed by the two contact positions. Suppose the two parallel fingers are  $f_1$  and  $f_2$ , the contact pair is  $[contact_a, contact_b]$ , the algorithm attaches  $f_1$  to  $contact_a$ , attaches  $f_2$  to  $contact_b$ , rotates the gripper around the axis  $contact_a - contact_b$ , and checks the collision between *eemesh* and *objmesh* at every orientation (see lines 13–18).

Fig. 8 shows an example of the planning algorithm using a Robotiq85 gripper and an electric drill. The CAD model of the drill is shown in Fig. 8(a). The collision between strokes and the model is detected in line 12 of Algorithm 2. Fig. 8(b.1–3) draw some results of the collision detection. The red cylinders

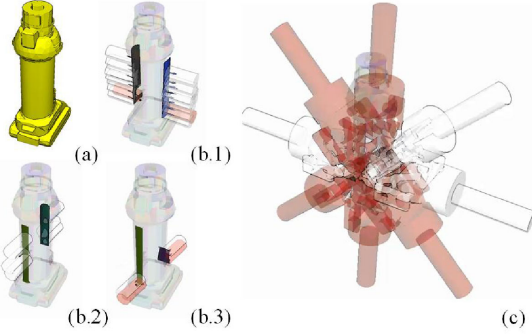


Fig. 8. Example of the grasp planning for a Robotiq85 gripper and an electric drill. (a) The CAD model of the drill. (b.1-3). Some of the parallel facets and the results of  $\text{checkCollision}(\text{stkMesh}, \text{objMesh})$ . The cylinders indicate the stroke of the gripper. The red ones collide with the drill. The white ones are collision-free. (c) The discretized grasps at one contact pair. The red hands indicate the obstructed grasps. The white ones are the planned grasps.

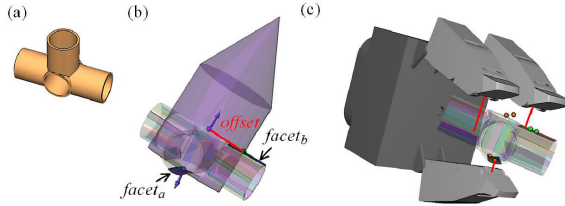


Fig. 9. Parameters used in the grasp planning for three-finger parallel grippers. (a) A clear view of the object used in (b) and (c). (b) The red segment indicates the thickness of the testing ray. It provides an offset between the thumb and the remaining fingertips. The center of the ray will be counted as  $\text{contact}_b$  if it intersects with  $\text{facet}_b$ . (c) The red segments indicate the distances between the two opposing fingers and the mesh. The planner requires the lengths of these segments to be smaller than  $t_{dct}$ .

show the strokes that collide with the object and are removed to avoid repeated collision checking at different orientations. The white cylinders are further examined in line 18 of Algorithm 2 to see if there is a collision between the whole hand and the object. The discretized orientations at one contact pair and the results of whole-hand collision detection are illustrated in Fig. 8(c). The red hands indicate the collided grasps found by  $\text{checkCollision}(\text{eemesh}, \text{objMesh})$ . The white ones are the planned grasps.

### C. Three-Finger Parallel Grippers

The planner proposed in this article is also applicable to three-finger parallel grippers, where two fingers are actuated together against a third finger.

The process is similar to grasp planning for two-finger parallel grippers, except that the parallel facets are treated as a heuristic guide. We only require the thumb finger to contact one of the parallel facets, while the two opposing fingers are allowed to touch anywhere of the object's mesh surface. The pseudocode is shown in Algorithm 3. In the first step, the planner finds parallel facets and computes candidate contact pairs using a routine similar to but more generalized than Algorithm 2. The main difference is we allow the thickness of a ray in the ray test to be changed, as is shown by the  $\text{offset}$  parameter of the  $\text{ray}()$  function in line 5. Algorithm 2 did not have this parameter, since

### Algorithm 3: Planning for Three-Finger Parallel Grippers.

```

1 candidatepairs  $\leftarrow$  []
2 for (faceta, facetb)  $\in$  objmesh do
3   if facetNormala·facetNormalb  $\leq$   $\cos(\theta_{\text{pari}})$ 
4     for (contacta, contactNormala)  $\in$  faceta do
5       ray  $\leftarrow$  ray(contacta, -contactNormala,
6         offset)
7       contactb  $\leftarrow$  intersect(ray, facetb)
8       if contactb is not None
9         candidatepairs.add([contacta,
10          contactb])
9 plannedgrasps  $\leftarrow$  []
10 for (contacta, contactb)  $\in$  candidatepairs do
11   for theta  $\in$  discreteangle do
12     /*size(discreteangles)  $\leftarrow$   $n_{da}$ */
13     tipPosition  $\leftarrow$  (contacta+contactb)/2)
14     rotmat  $\leftarrow$  rodrigues(contacta-contactb,
15       theta)
15     stkMesh.setPose(tipPosition, rotmat)
16     if not checkCollision(stkMesh, objMesh)
17       eemesh.setPose(tipPosition, rotmat)
18       if  $\text{dist}(f_2, \text{objMesh}) < t_{dct}$  and  $\text{dist}(f_3,$ 
19         objMesh)  $< t_{dct}$ 
20         if not checkCollision(eemesh,
21           objMesh)
22           if checkTorque(tipPosition,
23             objcom, hmax)
24             plannedgrasps.add([contacta,
25               contactb, rotmat])

```

it requires a pair of contacts to be confronting each other. In contrast, Algorithm 3 uses the parameter to allow the second contact to deviate from the piercing normal, and thus allows an offset between the thumb and the remaining fingertip contacts. Fig. 9(b) shows the details of how the  $\text{offset}$  works using the object shown in Fig. 9(a) as an example. Note that the  $\text{contact}_b$  in the algorithm is not a real contact on the object's mesh. It is the center of the thick ray at the collided cross section. In the second step, the planner poses the fingertips to the center of the contact pairs and performs collision detection at different rotations around the axis formed by  $\text{contact}_a$  and  $\text{contact}_b$ . The algorithm also invokes two nested  $\text{checkCollision}()$  functions. The first invocation checks the collision of strokes (see line 16). The second invocation checks the collision of the whole hand (see line 19). However, different from the two-finger case, the stroke of a three-finger gripper is represented by three cylinders, which changes with hand orientation. Thus, both the two invocations are performed under a specific orientation (both are invoked inside the loop initialized by line 11). If the stroke is collision-free, the planner attaches the single-finger tip ( $f_1$ ) to  $\text{contact}_a$  and the center of the other two fingertips ( $\frac{f_2+f_3}{2}$ ) to  $\text{contact}_b$  (see line 17), and examines the distances between  $f_2$  and  $\text{objMesh}$ , and  $f_3$  and  $\text{objMesh}$  (see line 18). The algorithm requires  $\|f_2 - \text{objMesh}\| < t_{dct}$  and  $\|f_3 - \text{objMesh}\| < t_{dct}$ , where  $t_{dct}$  is a tunable parameter. The parameter controls the contacts between the two opposing fingertips and the object surface. Note



TABLE I  
TUNABLE PARAMETERS

Name	Function	Where
$\theta_{pln}$	Control the planarity of each facet	Appeared in superimpose segmentation
$\theta_{fct}$	Control the overlap of facets	Appeared in superimpose segmentation
$t_{bdry}$	Control the distance between contacts and facet boundaries	Appeared in removing bad samples
$t_{rnn}$	Control the radius of contact regions	Appeared in removing bad samples
$h_{max}$	Control the stability of the planned grasps	Appeared in all grasp planners
$\theta_{parl}$	Control the parallelity of two facets	Appeared in the planners for two and three-finger grippers
$t_{dct}$	Control the distances between finger pads and object surfaces	Appeared in the planner for three-finger grippers
$n_{da}$	The number of discretized rotation angles around contact normals	Appeared in all grasp planners

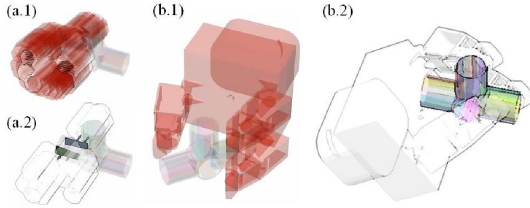


Fig. 10. Planning process for the three-finger gripper and the object shown in Fig. 9. (a.1). Results of the collision detection on strokes. (a.2) A clear view of the collision-free strokes of (a.1). (b.1) A collided grasp configuration found by the second collision checking. (b.2) A collision-free grasp configuration found by the planner.

that the algorithm does not constrain the contacts of  $f_2$  and  $f_3$  to fall in the parallel facet. The  $\text{dist}()$  function measures the distance to the object’s mesh surface. It allows the two fingers to contact anywhere on the mesh, as long as the distances are smaller than  $t_{dct}$ . After examining the distances, the algorithm checks the collision between the whole hand and the object (see line 19) and examines the stability of the grasping pose (see line 20).

Fig. 10 illustrates the planning process using the three-finger gripper and the object shown in Fig. 9. Fig. 10(a.1 and a.2) is the collision detection with strokes: Fig. 10(a.1) shows both the collided (red) and collision-free (white) strokes. The two-finger side is in the front. The strokes are rotated around the axis passing through the contact pair. Fig. 10(a.2) is a clear view without the collided ones. The collision-free strokes are further examined in the second collision detection. Some exemplary results are shown in Fig. 10(b.1) (a collided grasp configuration) and Fig. 10(b.2) (a collision-free grasp configuration).

## V. EXPERIMENTS, ANALYSIS, AND DEMONSTRATIONS

The experiments and analysis section has three parts. In the first part, we summarize the tunable parameters and analyze their influence on the planned results. In the second part, we study the computational costs, the robustness of the algorithm under different mesh qualities, the precision of the planned grasps, and the usage for mesh models reconstructed from point clouds. In the third part, we show some real-world systems that used the proposed method for grasp planning.

### A. Tunable Parameters

In the algorithms, seven tunable parameters are prepared for user configuration. The parameters and their functions are

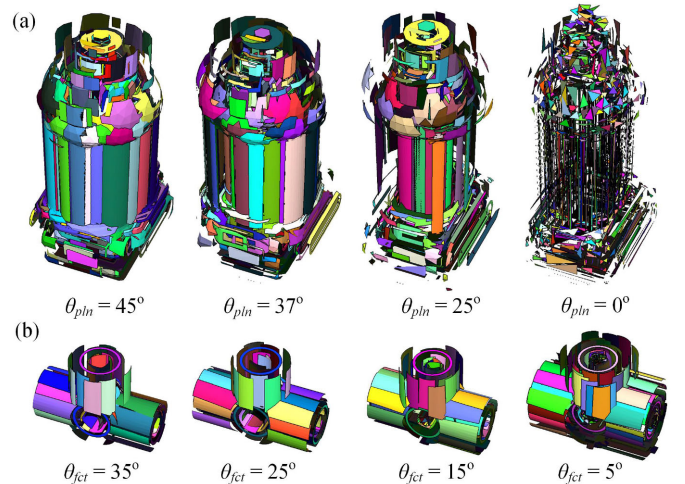


Fig. 11. (a) Results of superimposed segmentation using different  $\theta_{pln}$ . Each facet is shown in a random color and is drawn with a random offset from its original position to give a clear view. The results show that a smaller  $\theta_{pln}$  leads to flatter and smaller facets. (b) Results of superimposed segmentation using different  $\theta_{fct}$ . The object used for demonstration is the one in Fig. 9(a). The results show that a smaller  $\theta_{fct}$  leads to more overlap.

summarized in Table I. This section analyzes the parameters and compares the performance of different parameter settings by comparing the different planned results. In practice, users may set the parameters according to the needs of their robotic systems.

*Parameter 1:*  $\theta_{pln}$  is used to control the planarity of each facet during the superimposed segmentation. Smaller  $\theta_{pln}$  leads to flatter and smaller facets. Fig. 11(a) shows the segmented results of the electric drill shown in Fig. 8(a) using different  $\theta_{pln}$ . The facets are drawn with random offsets from their original position to give a clear view. Each facet is given a random color. As  $\theta_{pln}$  becomes small, facets become flatter and smaller. In the extreme case, where  $\theta_{pln} = 0^\circ$ , each triangle is treated as a facet.

*Parameter 2:*  $\theta_{fct}$  is used to control the overlap of superimposed facets. Smaller  $\theta_{fct}$  leads to more overlap between facets. Fig. 11(b) shows the segmented results of the tube connector shown in Fig. 9(a) using different  $\theta_{fct}$ . Like Fig. 11(a), each facet is drawn with a random offset and a random color. As  $\theta_{fct}$  becomes smaller, facets become more overlapped.

*Parameter 3:*  $t_{bdry}$  is used to control the distance between contacts and facet boundaries. Fig. 12(a) shows the results of contact sampling using the metal workpiece shown in Fig. 7(a) and different  $t_{bdry}$ . Only samples on the bottom of the object

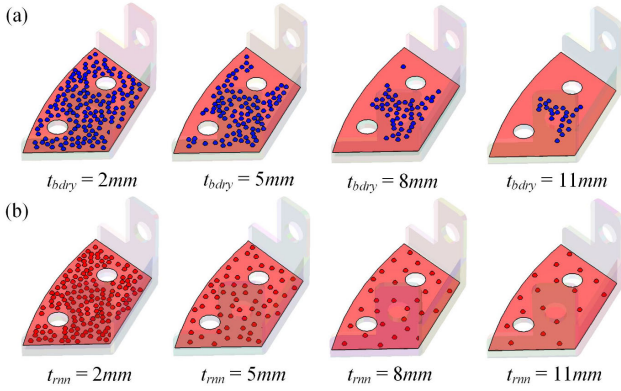


Fig. 12. (a) Results of contact sampling using different  $t_{bdry}$ . The resulting samples are drawn in blue color. The results show that a smaller  $t_{bdry}$  leads to a smaller clearance between contact samples and facet boundaries. (b) Results of contact sampling using different  $t_{rmn}$ . The resulting samples are drawn in red color. The results show that a smaller  $t_{rmn}$  leads to denser contact samples.

are shown, and the resulting samples are rendered in blue color. A smaller  $t_{bdry}$  leads to a smaller clearance between contact samples and facet boundaries (hence less robust results).

**Parameter 4:**  $t_{rmn}$  is used to control the radius of contact regions or the density of contact sampling. Fig. 12(b) shows the sampling results of the metal workpiece using different  $t_{rmn}$ . The samples are rendered in red. A smaller  $t_{rmn}$  leads to denser contact samples on the object surface (hence more planned grasps).

**Parameter 5:**  $h_{max}$  is used to control the stability of the planned grasps. An example is shown in Fig. 14(a). The object is the Stanford bunny (the last model in Fig. 15). As  $h_{max}$  decreases, the planner reduces to grasp flat facets near the center of mass to maintain stability.

**Parameter 6:**  $\theta_{parl}$  is used to control the parallelity of the facet pairs in grasp planning. Values with a larger offset from  $180^\circ$  lead to more candidate “parallel” facet pairs to attach the finger pads and hence more planned grasps. On the other hand, values with larger offsets result in unstable grasping configurations. Fig. 13 shows the parallel facets of a toy wheel using different  $\theta_{parl}$ . Arrows indicate facet normals. Two facets with the same arrow color are parallel. Like Fig. 11(a), the facets are drawn with random offsets from their original position to give a clear view. As  $\theta_{parl}$  becomes smaller, “parallel” facets become less parallel. Meanwhile, the number of parallel facets becomes larger.

**Parameter 7:**  $t_{dct}$  is used to control the distances between finger pads and object surfaces in the grasp planning for three-finger grippers. Larger values indicate the planner allows a large difference in distances between finger pads and object surfaces. In that case, there will be more planned grasps. Meanwhile, the results are less robust since two fingers cannot touch object surfaces simultaneously. Fig. 14(b) shows the planned grasps of the tube connector shown in Fig. 9(a) using different  $t_{dct}$ . As  $t_{dct}$  becomes larger, the planned grasp configurations become denser. The object, which is obstructed by hands in the figure, is at the same pose as Fig. 9(a). The results also show that when

$t_{dct}$  equals 0 mm, there are no lateral grasps. As  $t_{dct}$  becomes larger, the number of lateral grasps increases.

**Parameter 8:** In addition,  $n_{da}$  determines the number of discretized rotation angles around contact normals. A larger  $n_{da}$  leads to denser results, as is shown in Fig. 14(c).

## B. Performance

**1) Computational Costs:** The computational costs of planning grasps for various objects using the proposed algorithms are shown in Fig. 15. Eight objects are used. From left to right, they are as follows:

- 1) a bearing housing (bh);
- 2) a toy wheel (tw);
- 3) an electric drill (ed);
- 4) a tube connector (tc);
- 5) a metal workpiece (mw);
- 6) a toy plane body (pb);
- 7) a toy plane tail (pt);
- 8) the Stanford bunny (bny).

The details of these objects’ mesh models, including the number of vertices and triangle faces, are shown in the upper part of Fig. 15. The various computational costs, including the time spent on superimposed segmentation, sampling, removing bad samples 1 (refinement 1 of Fig. 4), removing bad samples 2 (refinement 2 of Fig. 4), planning contact pairs, and the two nested collision detection, are shown in a lower section of the table in Fig. 15. The results are obtained by running the algorithms on a LENOVO ThinkPad P70 mobile workstation. One core of an Intel Xeon E3-1505 M v5 @ 2.80 GHz 4 Core CPU is used. The memory size is 16.0 GB. The algorithms are implemented using python 2.7.11 32 b. The results are the average values of ten executions using the following parameter settings:  $\theta_{pln} = 20^\circ$ ,  $\theta_{fct} = 20^\circ$ ,  $t_{bdry} = 2$  mm,  $t_{rmn} = 3$  mm,  $h_{max} = 1.5$  mm,  $\theta_{parl} = 160^\circ$ ,  $t_{dct} = 3$  mm,  $n_{da} = 8$ .

The top two time-consuming items are marked by red and blue shadows in Fig. 15. The first one is `checkcollision(eemesh, objmesh)`, which required a few seconds for a few thousand triangles. The results are reasonable as we are performing mesh-to-mesh collision detections. The second one is “Remove Bad Samples 1” (see Section III.B.2) and Fig. 4). The cost is also reasonable, since it measures the distances of each sampled contact point to the boundaries of facets.

**2) Influence of Mesh Qualities:** Fig. 16 shows the performance of the algorithms using a model with different mesh qualities. The meshes are drawn in Fig. 16(1–8). The number of vertices and triangles of the meshes decreases from 1 to 8. The two charts in the figure show the normalized time costs and the normalized number of planned grasps on these models. The maximum values shown on the left side of the charts are used as denominators for normalization.

In detail, the different curves in Fig. 16 Chart(a) show the changes of various costs. The meanings of the colors are also shown on the left side of the charts. All the results are obtained using the same parameter settings:  $\theta_{pln} = 20^\circ$ ,  $\theta_{fct} = 20^\circ$ ,  $t_{bdry} = 2$  mm,  $t_{rmn} = 3$  mm,  $h_{max} = 1.5$  mm,  $\theta_{parl} = 160^\circ$ ,  $t_{dct} = 3$  mm, and  $n_{da} = 8$ .

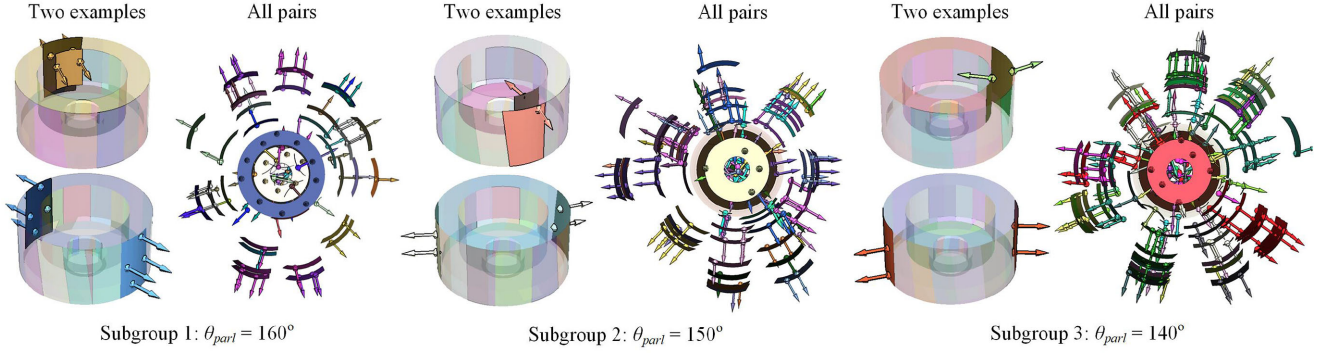


Fig. 13. Results of parallel facets using different  $\theta_{\text{parl}}$ . Arrows indicate facet normals. Two facets with the same arrow color are parallel. The left part of each subgroup shows two exemplary pairs. The results show that as  $\theta_{\text{parl}}$  decreases, less parallel facets are accepted. The right part of a subgroup shows all pairs. Here, the facets are drawn with random offsets from their original position to give a good view. As  $\theta_{\text{parl}}$  decreases, more pairs are found.

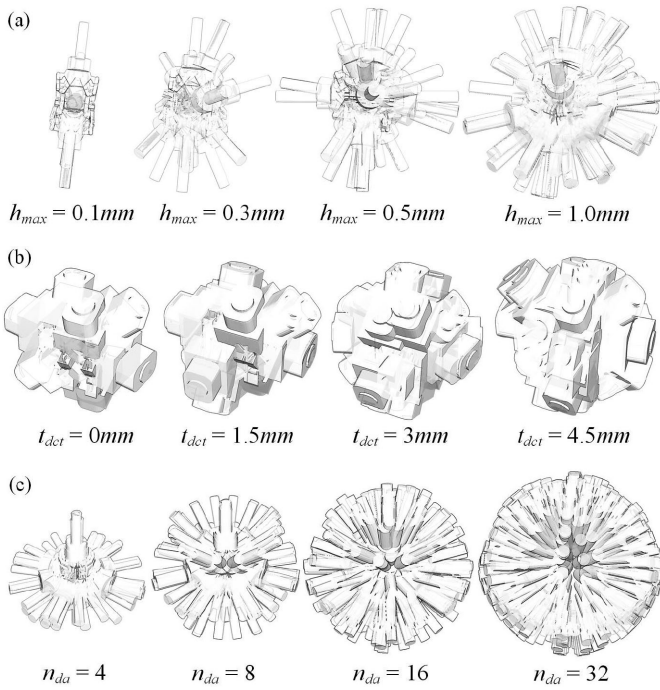


Fig. 14. (a) Results of grasp planning using different  $h_{\text{max}}$ . As  $h_{\text{max}}$  decreases, the hand grasps flat facets near the center of mass to maintain stability. Object: Stanford bunny, the last model in Fig. 15. (b) Results of different  $t_{\text{det}}$ . As  $t_{\text{det}}$  increases, more grasps are found. Object: Tube connector, the fourth model in Fig. 15. (c) Results of different  $n_{\text{da}}$ . A larger  $n_{\text{da}}$  leads to denser results. Object: Tube connector.

The two curves in Fig. 16 Chart (b) show the number of planned grasps. The red curve is the changes in grasp number using the same parameter setting. The blue curve shows the results using a different  $\theta_{\text{parl}}$  value ( $\theta_{\text{parl}} = 140^\circ$ ). The two curves imply that the algorithms are stable to low-quality mesh models: the number of planned grasps does not significantly decrease along with reduced vertices and triangles. For  $\theta_{\text{parl}} = 160^\circ$ , the number of planned grasps is considered to have similar values in the red shadow (spans from 1 to 4). For  $\theta_{\text{parl}} = 140^\circ$ , the number of planned grasps is considered to have similar values in the blue shadow (spans from 1 to 6). Note that the previous analysis is based on the single pt object. Further studies about other meshes

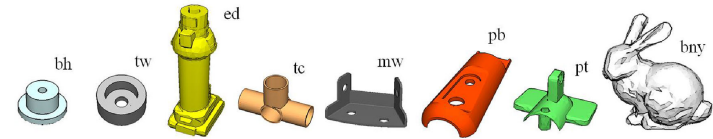
and meshes reconstructed from point clouds will be presented in Section V.B.3.

3) *Precision of the Planned Grasps*: Our goal is twofold in measuring the precision of the planned grasps. First, for a given mesh model, we measure the precision of the planned grasps under different parameter settings and study the influence of each parameter. Second, we measure the precision of the planned grasps under different mesh qualities and study the robustness of the proposed algorithms.

*Different Parameters*: We study the precision of the planned grasps under different parameter settings using a Robotiq85 two-finger parallel gripper. The experiment settings are shown in the right part of Fig. 17. The objects used are Fig. 17(a) the Stanford bunny (bny) and Fig. 17(b) the bearing housing (bh). The CAD models of these objects have fixed qualities: (#vertices 1392, #faces 2780) for bny; (#vertices 362, #faces 724) for bh. augmented reality (AR) markers are attached to the objects to precisely detect the changes of poses before and after closing the fingers. The grasps with approaching directions that have less than a  $40^\circ$  angle from the vertical direction, as shown in the left part of Fig. 17, are selected as the candidate grasps. They are used to grasp the objects. The initial object pose is considered to be a reference pose. The difference in object poses (the difference in the AR marker's  $x$  and  $y$  positions, namely  $d_x$  and  $d_y$ ) after closing the fingers is measured as the precision.

The results of the bny object are shown in Fig. 18. In the left case,  $\theta_{\text{parl}}$  was set to  $140^\circ$ , and six candidate grasps were found. The maximum change after grasping using these planned grasps was  $d_x = 1.00$  mm and  $d_y = 4.5$  mm. When  $\theta_{\text{parl}}$  was changed to  $160^\circ$  (the right part of Fig. 18), only one candidate grasp was found. Its change was  $d_x = 0.25$  mm and  $d_y = 1.00$  mm. The results of the bh object are shown in Fig. 19. The first two rows are the results using the following parameter settings:  $\theta_{\text{pln}} = 20^\circ$ ,  $\theta_{\text{fct}} = 20^\circ$ ,  $t_{\text{bdry}} = 2$  mm,  $t_{\text{mn}} = 3$  mm,  $h_{\text{max}} = 1.5$  mm,  $\theta_{\text{parl}} = 140^\circ$ ,  $t_{\text{det}} = 3$  mm, and  $n_{\text{da}} = 8$ . The lower two rows are the results using a different  $\theta_{\text{parl}}$  ( $\theta_{\text{parl}} = 160^\circ$ ). When  $\theta_{\text{parl}} = 140^\circ$ , there are 44 candidate grasps. Two failures are encountered during the experiments using these planned grasps. The failure cases are marked in red shadows. The reason was the object slipped out of the gripper owing to the low  $\theta_{\text{parl}}$ . When  $\theta_{\text{parl}} = 160^\circ$ , there are 32 candidate grasps, and all of them can successfully





Information of mesh models	#vertices	362	468	1225	816	316	1390	893	1392
#triangles		724	936	2441	1640	644	2786	1794	2780
Computational costs	Superimposed segmentation	0.154s	0.196s	0.936s	0.421s	0.137s	0.926s	0.509s	1.025s
	Sampling	0.050s	0.053s	0.095s	0.047s	0.038s	0.061s	0.067s	0.105s
	Remove bad samples 1	0.875s	1.281s	4.390s	1.993s	0.670s	6.805s	2.876s	5.586s
	Remove bad samples 2	0.020s	0.024s	0.063s	0.037s	0.012s	0.040s	0.025s	0.037s
	Plan contact pairs	0.144s	0.238s	1.634s	0.470s	0.051s	0.700s	0.208s	0.478s
	checkcollision( <i>stmesh</i> , <i>objmesh</i> )	0.236s	0.445s	0.142s	0.587s	0.104s	0.174s	0.179s	0.020s
	checkcollision( <i>emesh</i> , <i>objmesh</i> )	2.445s	2.917s	5.691s	2.899s	2.843s	2.286s	9.091s	1.175s

Fig. 15. Computational costs of the proposed algorithms. The rows marked by red and blue shadows denote the most time consuming process. The values are the average results of ten executions using the following parameter setting:  $\theta_{pln} = 20^\circ$ ,  $\theta_{fct} = 20^\circ$ ,  $t_{bdry} = 2$  mm,  $t_{rnn} = 3$  mm,  $t_{rss} = 50$  mm,  $\theta_{parl} = 160^\circ$ ,  $t_{dct} = 3$  mm,  $n_{da} = 8$ .

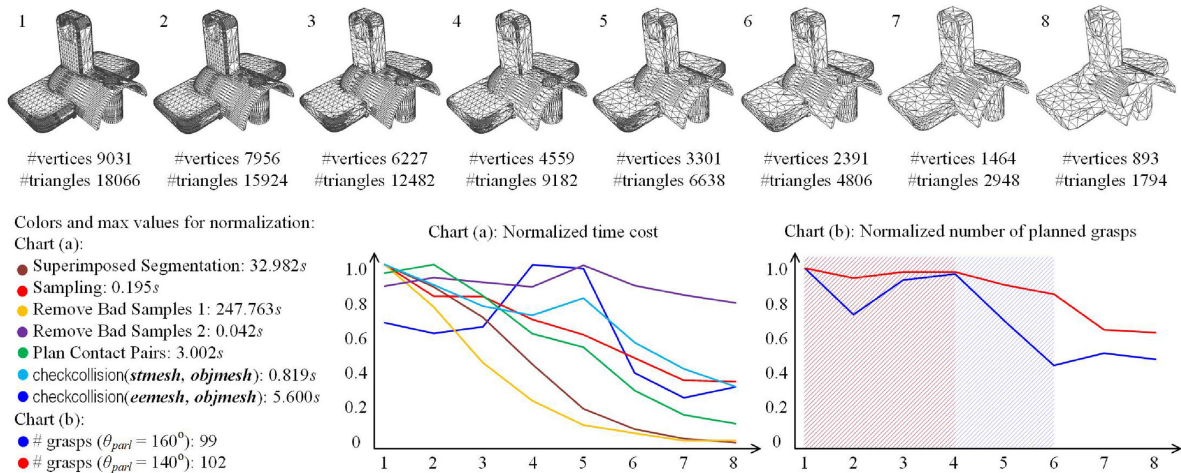


Fig. 16. Performance of the proposed algorithms using a model with decreasing mesh qualities. The meshes in 1–8 have a decreasing number of vertices and triangles, thus have declining quality. The curves in the two charts show the changes in time costs and the number of planned grasps as the mesh quality decreases.

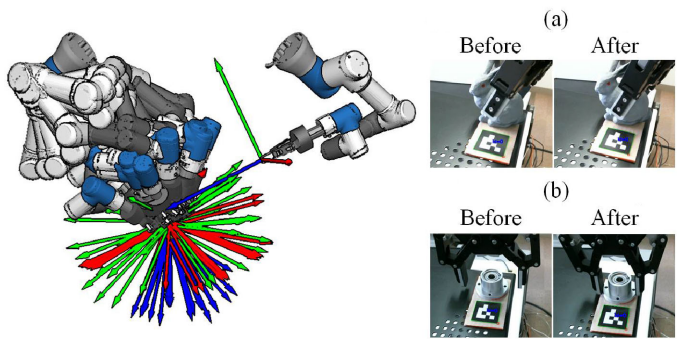


Fig. 17. Experimental settings used to examine the precision of the planned grasps. The left part shows the grasps with approaching directions that have less than a  $40^\circ$  angle from the vertical direction. The changes in object poses before and after grasping are measured by AR markers shown in the right. The bny and bh objects are tested.

hold the object. The maximum position changes after grasping are ( $d_x = 2.19$  mm,  $d_y = 1.90$  mm), and ( $d_x = 1.95$  mm,  $d_y = 1.88$  mm) in the two cases, respectively. They are marked in red

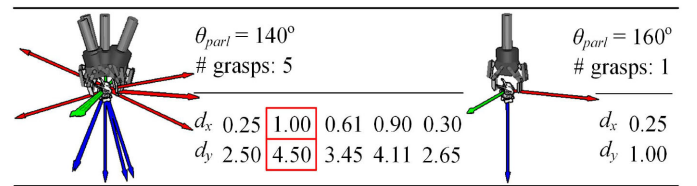


Fig. 18. Changes in positions after grasping the bny object using the planned grasps. The small 3-D figures show the candidate grasps used for comparison. When  $\theta_{parl} = 140^\circ$ , six candidate grasps are found. They are shown in the left 3-D figure. When  $\theta_{parl} = 160^\circ$ , only one candidate grasp is found. It is shown in the right 3-D figure.

frames. With the results of these two models, we confirm that the planned grasps have satisfying precision and are suitable to be used by assembly routines (e.g., spiral search). Also, we may change the algorithms' parameters and seek a balance between precision and the number of planned grasps according to the requirements of specific tasks (e.g., picking vs. assembly).

For readers' convenience, we summarize the detailed changes of segmentation and planned grasps of the bny object with

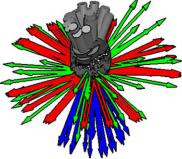
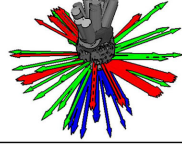
$\theta_{parl} = 140^\circ$ # grasps: 44		$d_x$	1.25	0.50	0.00	2.00	77.00	1.75	0.63	0.25	1.63	0.37	2.00	2.11	1.68	2.19	1.86	1.59
		$d_y$	1.63	0.63	0.00	0.00	17.63	0.75	0.63	0.00	0.63	0.30	0.10	0.21	1.03	0.57	0.40	0.96
		$d_x$	1.44	0.42	0.26	0.57	0.03	0.28	0.44	0.52	0.61	79.00	0.52	0.03	0.55	0.24	1.70	1.33
		$d_y$	1.14	0.35	0.28	0.24	0.30	0.20	0.45	0.50	0.84	68.10	0.83	0.18	0.24	0.04	0.86	1.27
		$d_x$	1.67	1.55	1.62	1.89	1.15	2.06	1.49	1.96	0.68	0.38	0.71	0.52	-	-	-	-
		$d_y$	0.60	1.13	0.74	0.50	1.90	0.35	1.32	0.29	0.79	0.47	0.95	0.49	-	-	-	-
$\theta_{parl} = 160^\circ$ # grasps: 32		$d_x$	0.25	0.75	0.25	0.00	1.75	0.00	0.00	0.61	0.64	0.00	0.77	1.95	0.59	0.73	1.17	1.90
		$d_y$	1.37	1.63	1.88	0.00	0.00	0.00	0.00	1.62	1.92	0.78	1.50	1.16	0.00	0.00	0.01	0.00
		$d_x$	0.97	1.12	0.28	0.86	0.43	0.82	0.00	0.06	0.25	1.25	0.28	0.09	0.12	0.94	0.53	1.03
		$d_y$	0.08	0.37	0.01	0.27	1.43	1.17	1.47	1.02	0.00	0.00	0.23	1.27	1.39	1.26	0.00	0.00

Fig. 19. Changes in  $x$  and  $y$  after grasping the bh object using planned grasps. When  $\theta_{parl} = 140^\circ$ , 44 candidate grasps were found. Their poses and precision are shown in the upper section of the table. The maximum positions change after grasping were  $d_x = 2.19$  mm and  $d_y = 1.90$  mm. They are marked using red frames. Two failures were encountered when grasping using these planned grasps. They are marked using red shadows. When  $\theta_{parl} = 160^\circ$ , 32 candidate grasps were found. Their poses and precision are shown in the lower section. The maximum changes were  $d_x = 1.95$  mm and  $d_y = 1.88$  mm.

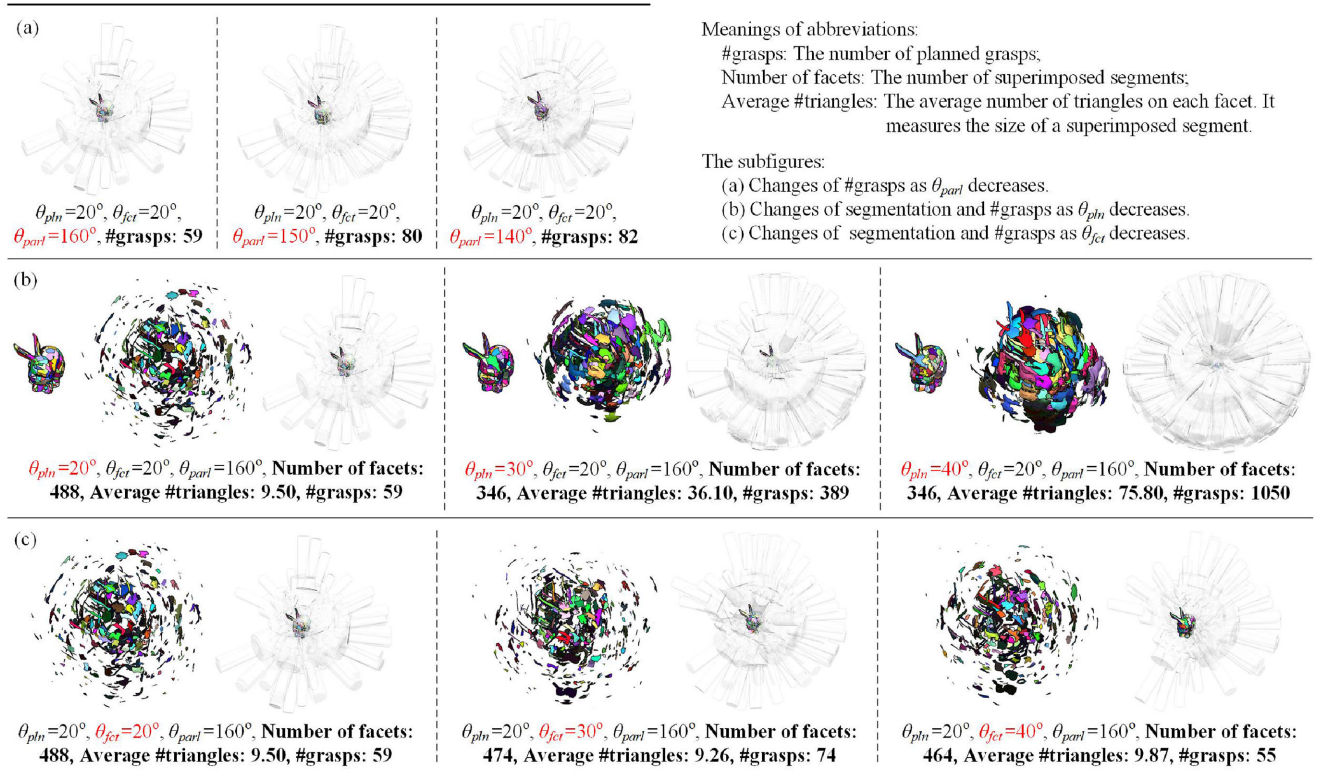


Fig. 20. Changes of segmentation and planned grasps of the bny object with different parameter settings. The parameters included  $\theta_{pln}$ ,  $\theta_{fct}$ , and  $\theta_{parl}$ . At each row, two of these parameters are fixed to  $160^\circ$ . The left parameter decreases from left to right.

different parameter settings in Fig. 20. In Fig. 20(a),  $\theta_{pln}$  and  $\theta_{fct}$  are fixed to  $20^\circ$ ,  $\theta_{parl}$  decreases from left to right. The number of planned grasps increases as  $\theta_{parl}$  decreases. In Fig. 20(b),  $\theta_{fct}$  is fixed to  $20^\circ$  and  $\theta_{parl}$  is fixed to  $160^\circ$ ,  $\theta_{pln}$  decreases from left to right. The sizes of facets (measured by the average number of triangles) increase significantly as  $\theta_{pln}$  decreases. Meanwhile, as the sizes of facets increase, the number of planned grasps increases significantly. In Fig. 20(c),  $\theta_{pln}$  is fixed to  $20^\circ$  and  $\theta_{parl}$  is fixed to  $160^\circ$ ,  $\theta_{fct}$  decreases from left to right. The number of facets decreases slightly as  $\theta_{fct}$  gets smaller. There

are no necessary relations between  $\theta_{fct}$  and the sizes of facets, or between  $\theta_{fct}$  and the number of planned grasps.

*Different Mesh Qualities.* We measure the precision of the planned grasps under different mesh qualities by using both a Robotiq85 two-finger parallel gripper and a suction cup. The method we perform the measurement is to plan grasps using low-quality mesh models, and compare the contact normals of the planned grasps with the real surface normal on a groundtruth. Here, the model with the best quality (original one modeled using an industrial CAD software, i.e., PTC



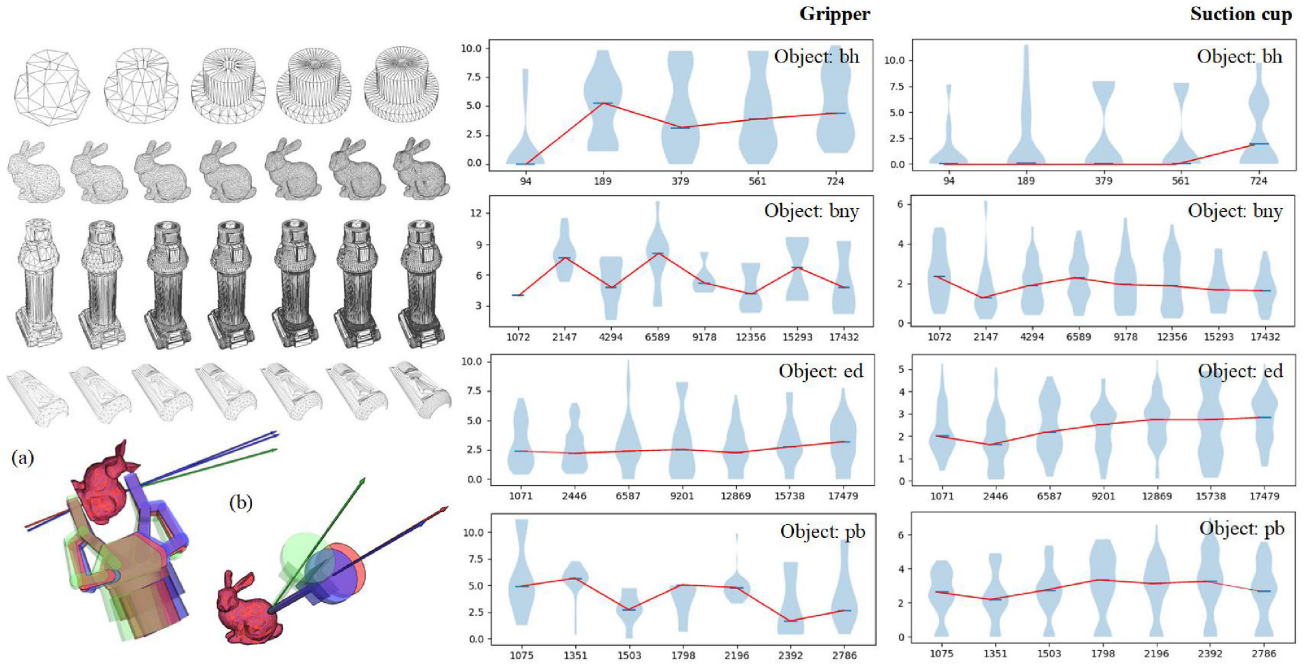


Fig. 21. Changes of planned grasp/suction precision under different mesh qualities. The four objects with different mesh qualities shown in the upper-left part are used to plan grasps and suction. The precisions of the planned results are plotted in the middle (gripper) and right (suction cup) columns. The horizontal axis of each plot indicates the number of triangles in the mesh models in the left. The vertical axis is the precision (the angular difference represented in degree).

Creo Parametric) is used as the groundtruth. The precision of a planned grasp is measured as the difference between the contact normal on the low-quality mesh and the groundtruth mesh (the in-between angle). Mathematically, the precision is represented using  $p_f = \arccos(\hat{n}_0 \cdot n_0) + \arccos(\hat{n}_1 \cdot n_1)$  for the parallel gripper, which has two contact normals represented by  $n_0$  and  $n_1$ , and  $p_f = \arccos(\hat{n} \cdot n)$  for the suction cup, which has only one contact normal represented by  $n$ .

The results are shown in Fig. 21. Four objects (bh, bny, ed, and pb) with changing mesh qualities are used in the analysis, for they have both spherical and flat features and are easier to compare. The left part of the figure illustrates the meshes. The right part shows the changes in the planned grasp precisions. In detail, the right part has two columns. The charts in the first column are the results of a parallel gripper. The charts in the second column are the results of a suction cup. The parameters used in the planner are the same as in Section V.B.1. Note that the charts are using absolute values. Their horizontal axes sequentially indicate the number of triangles in each mesh on the left. Their vertical axes are the angular difference (in degree).

The results indicate that for both the gripper and the suction cup, our algorithms can plan grasps with an angle difference of less than  $\pm 12^\circ$ . It is consistent with our parameter setting ( $\theta_{\text{part}} = 160^\circ$ ,  $\theta_{\text{pln}} = 20^\circ$ ). The results also show that lower mesh qualities do not decrease the precision, except in the extreme cases of bh and pb. The lower-left corner illustrates two examples of the planned grasps/suctions for the readers' convenience. In Fig. 21(a), the red one is a planned grasp using the most low-quality bny model. The blue one is a grasp at the same contact but with a normal from the groundtruth mesh. The green one is also a grasp at the same contact, but the normal is from a triangle, not a

facet. The blue and green grasps can be considered as the results of ray-shooting based grasp planning (see next section). The figure shows that the red one is nearer to the blue one compared to the green one, which indicates the grasp planned by our planner improves the precision. Fig. 21(b) shows three suction cups. The red, blue, and green colors indicate the same planned results. Likewise, the red one is nearer to the blue one and has higher precision. The results indicate that our method has high precision and is less influenced by different mesh qualities.

4) *Comparison With Other Methods:* The developed algorithms are compared to two other widely used grasp planning methods. The first one is the ray-shooting method, which samples a contact point on the mesh surface and finds the candidate counter contact point by shooting a ray from the sampled point along the reversed normal direction. When the angle between the normal at the intersection and the normal of the sampled contact point is less than  $\theta_{\text{part}}$ , the sampled contact point and the candidate counter contact point are counted as a gripping pair. The second one is a similar segmentation and sampling-based method. The difference is the superimposition between the facets is not considered. It is thus called a simple segmentation method.

The results of these methods using the mw and tc objects as examples are shown in Fig. 22(a). For each object, three rows are used to show the results of the ray-shooting method, the simple segmentation method, and the proposed method. The proposed method finds more grasps (2080 vs. 140/1984 for the mw object and 5056 vs. 368/2848 for the tc object). Meanwhile, it has no significantly higher time cost (2.834s vs. 2.142/2.073s for the mw object and 13.271s vs. 8.476/6.326s for the tc object).

More statistical results are presented in Fig. 22(b) and (c) and Fig. 23. Fig. 22(b) compares the total time costs, the number



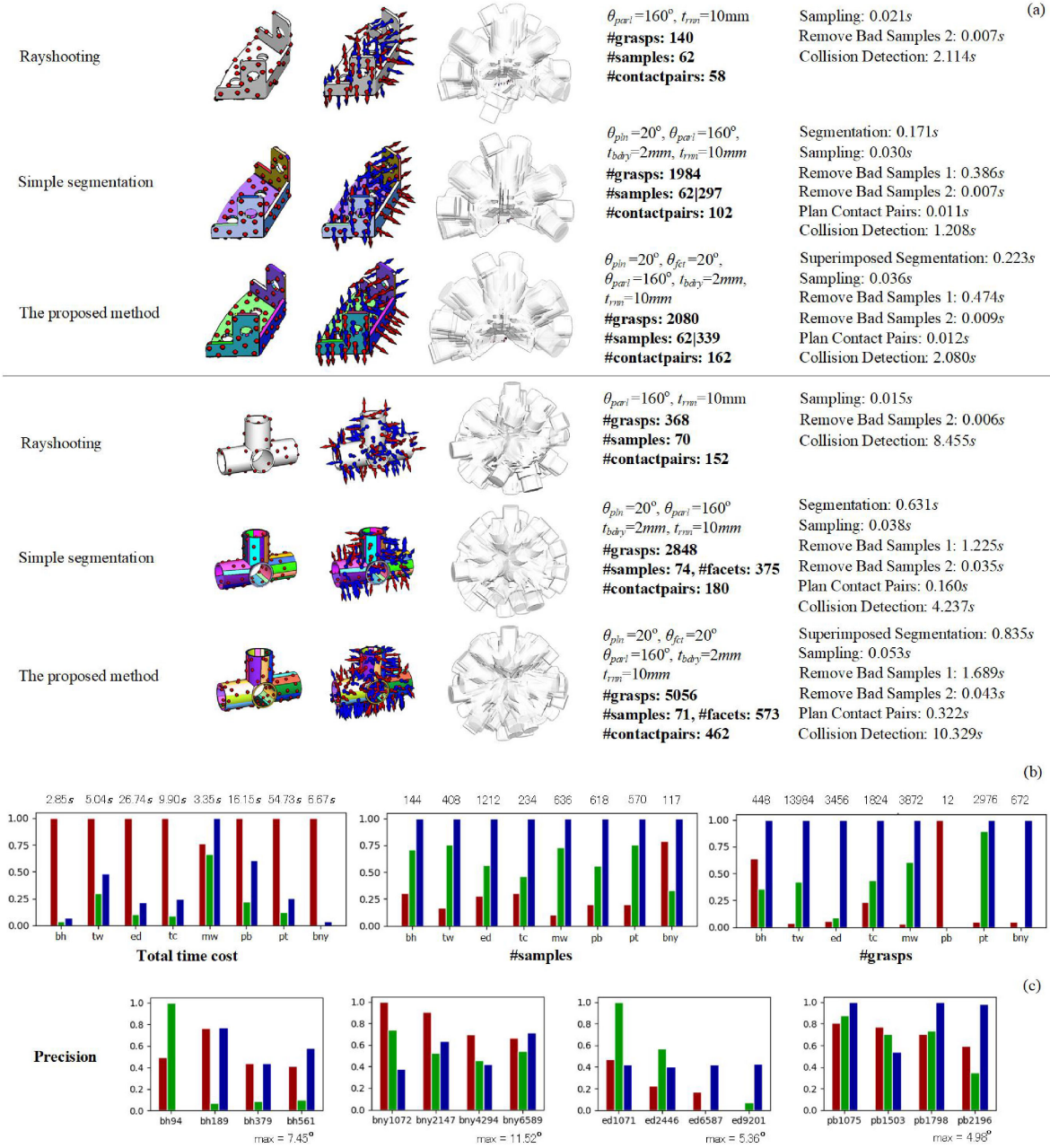


Fig. 22. Comparing the performance of a rayshooting method, a simple segmentation method, and the proposed method. (a) The results on the mw and tc objects. Graphical illustrations are shown in detail for these two objects. (b) Statistical results over all objects in Fig. 15. The three bar graphs compare the total time costs, the number of sampled points, and the number of grasps of the three methods. The red, green, and blue colors correspond to the results of the rayshooting, the simple segmentation, and the proposed method, respectively. The horizontal axis indicates the different objects. The vertical axes are normalized concerning the maximum values of the corresponding objects, which are shown on the top of the graphs for readers' reference. (c) Comparing the precision of the three methods using the four objects with varying mesh qualities shown in Fig. 21. The colors have the same meaning as (b). The horizontal axis indicates the number of triangles in the mesh. The vertical axis indicates the normalized average angular difference. The maximum difference values used for normalization are shown in the lower right corners of each graph for readers' reference.

of sampled points, and the number of planned grasps of the three methods using all objects in Fig. 15. Fig. 22(c) compares the precision of the three methods using the four objects with varying mesh qualities shown in Fig. 21. The red, green, and blue bars of these two subfigures indicate the ray-shooting method, the simple segmentation method, and the proposed method, respectively. The horizontal axis indicates the different objects or an object with different mesh qualities. The vertical axis is normalized concerning the maximum value of the corresponding

object. The results show that the proposed method generally finds more samples and grasps. Its time cost is less than the ray-shooting method and is not significantly worse than the simple segmentation method. Its precision is less affected by mesh qualities.

Fig. 23 further compares the sizes of the facets. Likewise, all eight objects in Fig. 15 are used, and the results are summarized in the eight plots. The vertical axes of these graphs indicate the size of a facet. The horizontal axes indicate the IDs of

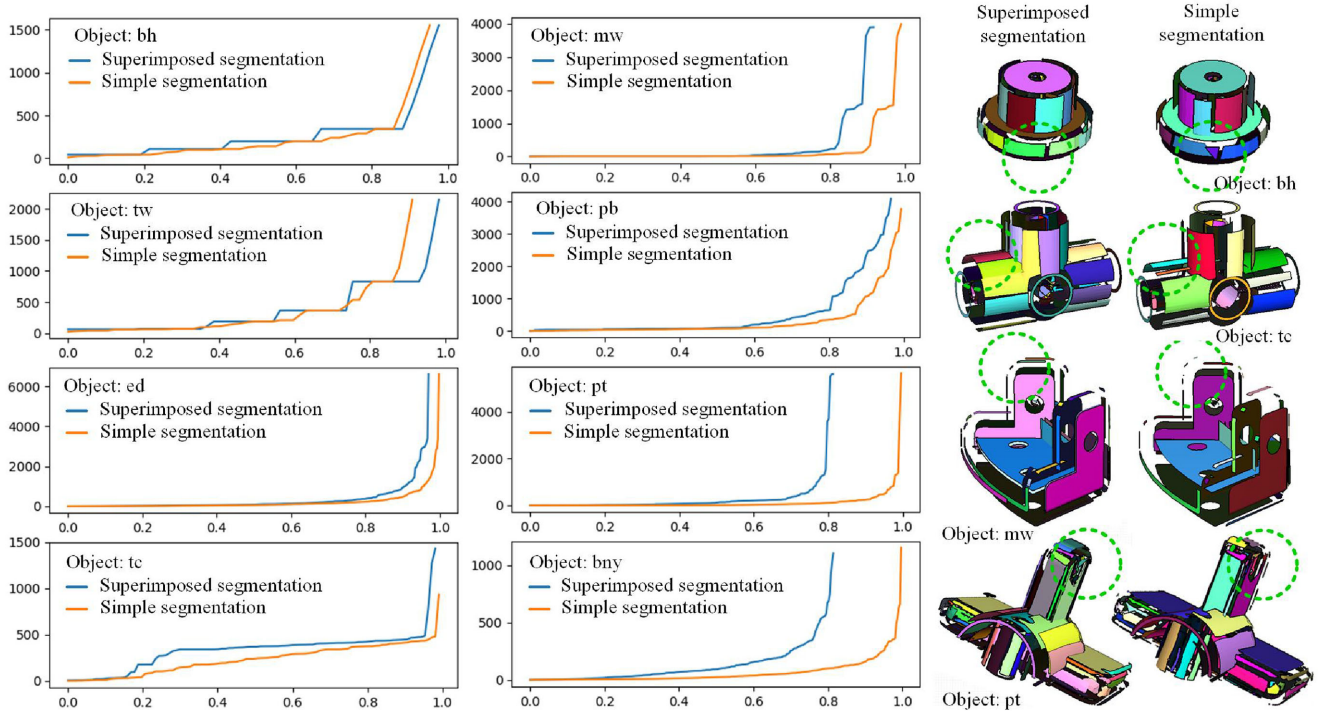


Fig. 23. Comparing the facet sizes of the method using simple segmentation and superimposed segmentation. The vertical axis of each plot indicates the size of a facet. The horizontal axes indicate the ID of a facet. The IDs are normalized to the maximum value to have a better visual effect. The blue curve shows the facet sizes of the method using superimposed segmentation. The orange curve shows the sizes using simple segmentation. The right part renders the difference graphically for better inspection.

the facets. The IDs are normalized to the maximum value to have a better visual effect. In each graph, there are two curves, where the blue curve shows the sizes of the facets produced by superimposed segmentation, and the orange curve shows the sizes of simple segmentation. The IDs are sorted so that the vertical values increase monotonically. The curves imply that after superimposed segmentation, the sizes of the facets increased. Meanwhile, instead of smoothly increasing, the sizes change by steps. See the first graph, for example. Before segmentation, the orange curve is increasing smoothly. After segmentation, the blue curve changes into a multistep function. The small facets are merged into bigger ones that have similar curvature owing to the allowance of repetition, and the facets are better categorized according to their geometry features. The right part of this figure further shows the changes by rendering facets in 3-D. Here, four representative cases are shown, including bg, tc, mw, and pt. The left column shows the facets produced by superimposed segmentation. The right column shows the results of simple segmentation. The area marked by the dashed circle denotes the merged little pieces. They disappeared in the left column. The results demonstrate that the proposed method is more uniform in segmentation and consequentially more robust in grasp planning.

5) *Applying to Point Clouds*: Since the proposed planner has low dependency on the mesh qualities, it is suitable for planning grasps using the mesh models reconstructed from point clouds obtained by depth sensors. This section demonstrates and analyzes such an application. We use Photoneo Phoxi3D to capture point clouds, use the conventional ball pivoting algorithm [57]

to reconstruct a mesh model, and use the proposed planner to plan both suction and gripping poses. The suction cups have two different sizes. The first one is the same as the previous section. Its rubber pad diameter is 15 mm. The second suction cup is a larger one with a diameter of 45 mm. To avoid the bothering fixing work, we ask an ABB Yumi robot to hold the suction cups when executing the planned suction action.

*Suction Poses for a Piece of Point Cloud*. Planning suction poses for a piece of point cloud is simpler compared to that of a gripper, for there is no request for parallel facets. A piece of point cloud captured from a single view is enough to find candidate suction poses. Fig. 24 shows the preprocessing done to get the piece of point cloud and the reconstructed mesh model. The Phoxi3D is installed to the top of the scene shown in Fig. 24(a). It captures a point cloud shown in Fig. 24(b). Fig. 24(c) shows a close-up view of the captured points. Background subtraction is performed in Fig. 24(d) to crop out the point clouds that correspond to the objects, say object point clouds. These point clouds are clustered using density-based spatial clustering of applications with noise (DBSCAN) [58] in Fig. 24(e) to separate different objects. Each separated object point cloud is reconstructed into a mesh model in Fig. 24(f). The last row of this figure shows a close-up view of the left object to let readers better compare the results in the following experiments. The object is an alternative head of a Dyson vacuum cleaner. It is made of a long flat mouth and a cylindrical connecting end.

Fig. 25 shows the planned suction poses of the small suction cup using the reconstructed mesh models. In Fig. 25(a),  $\theta_{pln}$  and  $\theta_{fct}$  are fixed to  $20^\circ$ . The  $t_{brdy}$  parameter is set to be half of its



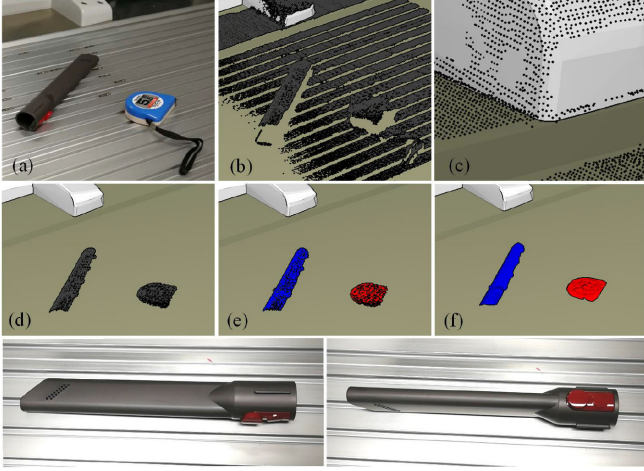


Fig. 24. Reconstructing the mesh model from a single view to prepare for planning the suction poses. (a) The real-world scene. (b) Captured point clouds. (c) A close-up view of the captured point clouds. (d) Background subtraction. (e) Cropping out the object point clouds. (f) Reconstructed mesh models. The flat last: A close-up view of the left object.

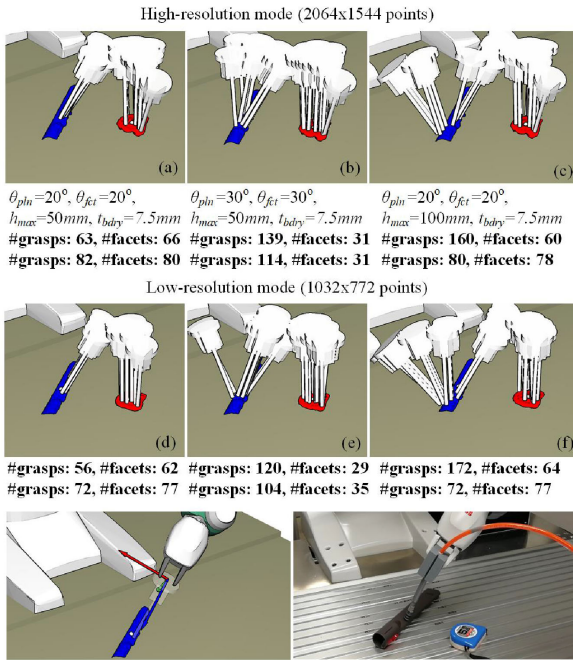
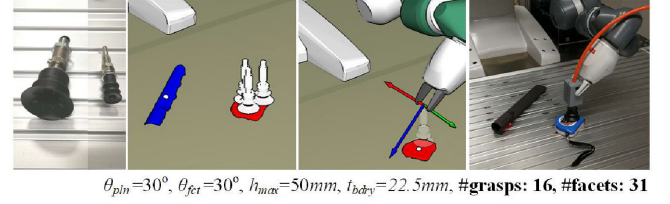


Fig. 25. Upper two rows: Planned suction poses using the reconstructed mesh models. (a), (b), and (c) are the results under a high resolution-mode. (d), (e), and (f) are the results under a low-resolution mode. The last row: A final suction pose chosen by a robot motion planner.

pad diameter (7.5 mm). Since the suction cup exerts an active sucking force, there is no need to consider SFC. Thus, (5) is degenerated into  $c \leq h_{max}$ —namely  $h_{max}$  controls the distance to the *com*. The  $h_{max}$  parameter in Fig. 25(a) is set to 50 mm. The white spheres on the two mesh models indicate their *coms*. In Fig. 25(b), both  $\theta_{pln}$  and  $\theta_{fct}$  are changed to  $30^\circ$ . The  $h_{max}$  parameter is kept the same. In Fig. 25(c),  $\theta_{pln}$  and  $\theta_{fct}$  are the same as Fig. 25(a), but the  $h_{max}$  parameter is set to 100 mm. By comparing the results in Fig. 25(a) and (b) we found that smaller  $\theta_{pln}$  and  $\theta_{fct}$  reduce the suction poses around the cylindrical



$\theta_{pln}=30^\circ, \theta_{fct}=30^\circ, h_{max}=50mm, t_{brdy}=22.5mm, \#grasps: 16, \#facets: 31$

Fig. 26. From left to right: The two suction cups; Planned suction poses; A final suction pose selected by the robot motion planner; Real execution.

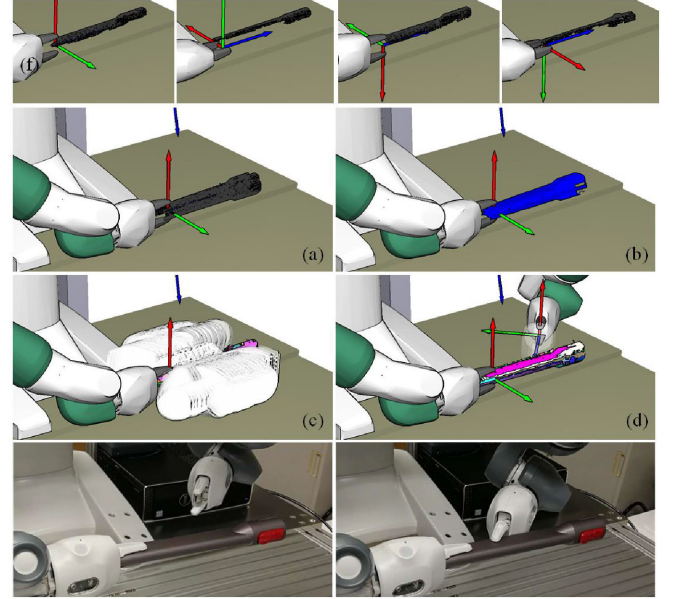


Fig. 27. Planning grasps poses for the ABB IRB14000 SmartGriper using point clouds. First row: Capture and merge point clouds from four views. (a) The merged point cloud. (b) The reconstructed mesh model. (c) The planned grasps. (d) A final grasp chosen by a robot motion planner. The last row: The real-robot execution.

connecting end of the object, thus resulting into a smaller number of planned results. By reviewing Fig. 25(c), we found the planner avoided the bumps at the connecting end of the object to ensure flat and large contact areas.

The Photoneo Phoxi3D has a high-resolution mode ( $2064 \times 1544$  points) and a low-resolution mode ( $1032 \times 772$  points). The results in Fig. 25(a) are under high resolution mode. We further carried out experiments to compare the performance of our method under the low-resolution mode. The results are shown in Fig. 25(d)–(f). The parameter settings of them are, respectively, the same as Fig. 25(a)–(c). The results indicate that there is no significant difference compared to the high-resolution mode. Note that the ball pivoting radius in these two modes are both set to 5 mm to reduce the influence of the mesh reconstruction method. The last row of this figure shows a final suction pose chosen by a robot motion planner and the real-robot execution.

Fig. 26 shows the results of the larger suction cup. In this case, the  $t_{brdy}$  parameter is set to be 22.5 mm. The Photoneo Phoxi3D is in low-resolution mode. The algorithms told us the cleaner head could not be sucked up by the large cup. The reason is no



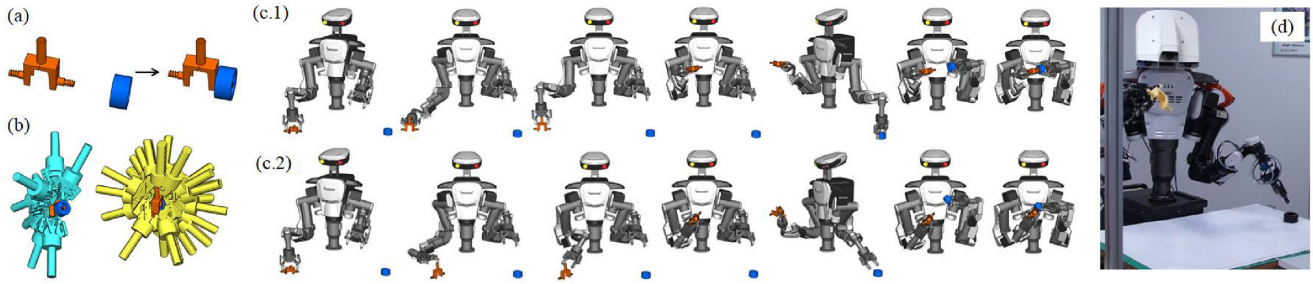


Fig. 28. Using a Kawada Nextage robot to conduct an assembly task. (a) The goal of the assembly task. (b) The planned collision-free grasps for assembly. The cyan hands show the possible grasp configurations to hold the support. The yellow hands show the possible grasp configurations to hold the wheel. (c) Two exemplary motion sequences using the planned grasps. (d) A real-world execution.

contact with a distance smaller than 22.5 mm to facet boundaries was found.

*Gripping Poses by Merging Several Point Clouds.* Planning grasping poses for a parallel gripper requires parallel facets. Thus, it is impossible to carry out experiments with the point clouds from a single view. We solve this by asking one arm of the ABB Yumi robot to hold the object, rotate it 360° with 90° step length to capture four point clouds, and merge them. This rotation-merging process is shown in the first row of Fig. 27. The merged point cloud is shown in Fig. 27(a). The ball pivoting algorithm is applied to reconstruct a mesh model in Fig. 27(b). The grasp planner plans grasps using the reconstructed model, and the results are shown in Fig. 27(c). Note that all the planned grasps grip at the long flat mouth of the object. Directly holding the connecting end is avoided. The reason is that the planner did not find good contacts near the bumps at the connecting end. Fig. 27(d) shows a final grasp chosen by a robot motion planner. Here, each facet of the reconstructed mesh model is drawn with different colors to give a clear view of the hidden object in Fig. 27(c). The last row shows the real-robot execution.

### C. Some Real-World Examples Using the Grasp Planner

The proposed method is implemented as a plugin of an open-source project named *wrs\_nedo* (available on Github at [https://gitlab.com/wanweiwei07/wrs\\_nedo](https://gitlab.com/wanweiwei07/wrs_nedo)), where the goal is to build a system that conducts assembly tasks without human teach. The input to the system is the CAD models of objects and kinematic parameters of robots and hands. The output is a sequence of grasp and robot configurations for assembly. The proposed grasp planning is a preprocessing component that prepares preannotated grasp poses. The preannotated grasps are used by a task and motion planning component to build a manipulation graph [46] for selecting grasps and generating robot motion. This section presents some real-world examples developed based on the system, with a special focus on grasp planning. Note that all parameters are set to the following default values during planning the grasps:  $\theta_{\text{pln}} = 20^\circ$ ,  $\theta_{\text{fct}} = 20^\circ$ ,  $t_{\text{bdry}} = 2 \text{ mm}$ ,  $t_{\text{rnn}} = 3 \text{ mm}$ ,  $h_{\text{max}} = 1.5 \text{ mm}$  (50 mm for a suction cup),  $\theta_{\text{parl}} = 160^\circ$ ,  $t_{\text{dct}} = 3 \text{ mm}$ ,  $n_{\text{da}} = 8$ .

The first example uses a Kawada Nextage robot (Kawada Industries, Inc.) to assemble a wheel to a support shown in Fig. 28(a). The hands are two Robotiq F-85 grippers. The

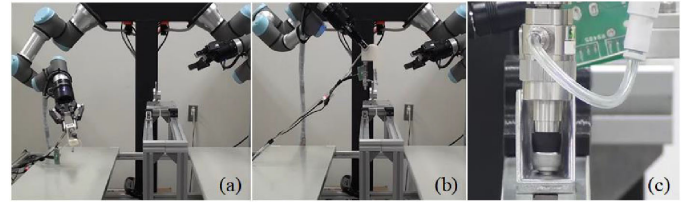


Fig. 29. Using a dual-arm UR3 robot to pick up a vacuum fastener and fasten a bolt. (a), (b) The robot chooses a suitable grasp from preplanned grasps to pick up the vacuum fastener. (c) The robot aligns the suction tooltip to a bolt and fastens it.

input is the initial positions and orientations of the wheel and the support in a robot’s workspace. The system automatically plans grasps, invalidates collided grasp configurations, and plans motion sequences using the collision-free grasps. The planned collision-free grasps for assembling the two objects are shown in Fig. 28(b). Two exemplary planned sequences are shown in Fig. 28(c.1) and (c.2). The real-world execution using one of the planned sequences is shown in Fig. 28(d).

The second example uses a dual-arm UR3 robot (Universal Robots A/S) to pick up a vacuum fastener and fasten a bolt. The robot plans a bunch of candidate grasps using the proposed algorithms and chooses a suitable grasp to pick up the vacuum fastener in Fig. 29(a) and (b). Then, it aligns the suction tooltip to a bolt and fastens it in Fig. 29(d). Like the first example, the hand used is a Robotiq F-85 gripper.

The third example is to hand over an electric drill from the right hand of an HRP5P robot (National Inst. of AIST, Japan, <http://y2u.be/ARpd5J5gDMk>) to its left hand. The hands are two three-finger parallel grippers. The robot could precisely hold the object using the planned grasps [see Fig. 30(a)], and conduct handover by inserting the thumb of the left hand into the narrow space between the index and middle fingers of the right hand Fig. 30(b) and (c).

The fourth example uses an ABB Yumi robot and its two integrated IRB14000 SmartGrippers to pick up the aforementioned cleaner head and drop it to a container. Each of the SmartGrippers has both a lateral suction cup and a central parallel gripper. The robot uses the right-hand suction cup to pick up the head [Fig. 31(a), the object is not grippable due to collisions with the table] and hands it over to the left-hand parallel gripper for dropping [Fig. 31(b) and (c)].

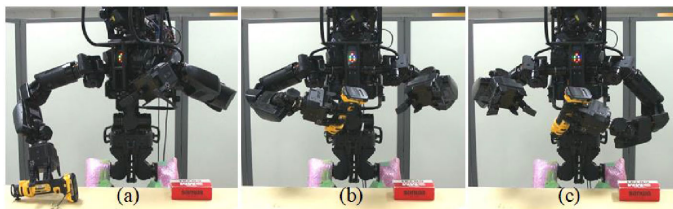


Fig. 30. Handing over a tool driver from the right hand of an HRP5P robot to its left hand. (a) The robot picks up the electric drill using its left hand and prepares it for handover. (b) and (c) The robot inserted the thumb of its right hand into the narrow space between the index and middle fingers of the left hand to hand over the drill.

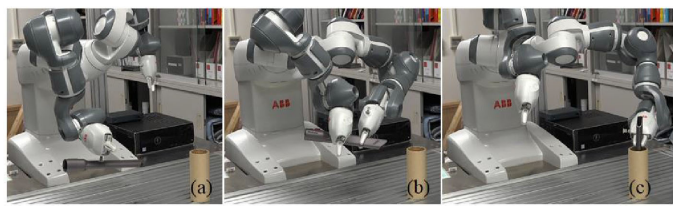


Fig. 31. Using combined suction cups and parallel grippers to pick up the aforementioned cleaner head and drop it to a container. (a) The robot picks the object up using its right-hand suction cup. (b) The robot hands over the object to its left-hand parallel gripper. (c) The robot drops the object to a container.

The details of these examples together with their grasp planning process are available in the video supplementary of this article.

## VI. CONCLUSION

In this article, we proposed grasp synthesis algorithms that are efficient, precise, and highly configurable with several tunable parameters. The algorithms focused on industrial end-effectors like grippers and suction cups, and can be used in industrial tasks like bin-picking and assembly. The efficiency, precision, robustness, and configurability of the proposed method meet the requirements of these tasks. The proposed planner was independent of mesh qualities and was applicable to mesh models reconstructed from point clouds. It was demonstrated to be practical by a real-world robotic assembly task.

We envision our future research focusing on noncontinuous contacts. Right now, a finger pad was assumed to be continuously in contact with an object, making it difficult to find grasps on discrete surfaces (e.g., grasp the screw threads of a bolt). We will use surface simplification, reconstruction, and multi-contact analysis to challenge the difficulty in the future.

## REFERENCES

- [1] J. K. Salisbury, "Kinematic and force analysis of articulated hands," in *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA, USA: MIT Press, 1985, pp. 2–167.
- [2] B. Mishra, J. T. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica*, vol. 2, pp. 541–558, 1987.
- [3] V.-D. Nguyen, "Constructing force-closure grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1986, pp. 1368–1373.
- [4] Y. Funahashi, T. Yamada, M. Tate, and Y. Suzuki, "Grasp stability analysis considering the curvatures at contact points," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 3040–3046.

- [5] E. Rimon and J. W. Burdick, "Mobility of bodies in contact – I: A new 2nd order mobility index for multiple-finger grasps," *IEEE Trans. Robot. Automat.*, vol. 14, no. 5, pp. 696–708, Oct. 1998.
- [6] Y.-H. Liu, "Computing n-finger form-closure grasps on polygonal objects," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 149–158, 2000.
- [7] A. F. van der Stappen, "Immobilization: Analysis, existence, and output-sensitive synthesis," in *Computer -Aided Des. and Manuf.*, American Mathematical Society (AMS), Rhode Island, 2005, pp. 162–187.
- [8] D. J. Montagna, "The condition for contact grasp stability," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, pp. 412–417.
- [9] A. Bicchi, "On the closure properties of grasping," *Int. J. Robot. Res.*, vol. 14, pp. 319–334, 1995.
- [10] W. S. Howard and V. Kumar, "On the stability of grasped objects," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 904–917, Dec. 1996.
- [11] B. Mishra, "Grasp metrics: Optimality and complexity," in *Proc. Workshop Algorithmic Found. Robot.*, 1995, pp. 137–165.
- [12] M. A. Roa and R. Suárez, "Grasp quality measures: Review and performance," *J. Auton. Robots*, vol. 38, pp. 65–88, 2015.
- [13] J. Ponce, S. Sullivan, A. Sudsang, and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *IEEE Trans. Robot. Autom.*, vol. 16, no. 16, pp. 11–35, Feb. 1997.
- [14] 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.
- [15] M. Fischer and G. Hirzinger, "Fast planning of precision grasps for 3D objects," in *Proc. IEEE/RJS Int. Conf. Intell. Robots Syst.*, 1997, pp. 120–126.
- [16] A. Miller and P. Allen, "Examples of 3D grasp quality computations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, pp. 1240–1246.
- [17] J. D. Wolter, R. A. Volz, and A. C. Woo, "Automatic generation of gripping positions," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 2, pp. 204–213, Mar./Apr. 1985.
- [18] J. L. Jones and T. Lozano-Perez, "Planning two-fingered grasps for pick-and-place operations on polyhedra," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 683–687.
- [19] C. Liu, H. Qiao, J. Su, and P. Zhang, "Vision-based 3-D grasping of 3-D objects with a simple 2-D gripper," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 605–620, May 2014.
- [20] M. Pozzi, M. Malvezzi, and D. Prattichizzo, "On grasp quality measures: Grasp robustness and contact force distribution in underactuated and compliant robotic hands," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 329–336, Jan. 2017.
- [21] J. Shi and G. S. Koonjul, "Real-time grasping planning for robotic bin-picking and kitting applications," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 809–819, Apr. 2017.
- [22] Y. Li, J. P. Saut, J. Pettre, A. Sahbani, and F. Multon, "Fast grasp planning using cord geometry," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1393–1403, Dec. 2015.
- [23] M. Ciocarlie, C. Lackner, and P. Allen, "Soft finger model with adaptive contact geometry for grasping and manipulation tasks," in *Proc. Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, 2007, pp. 219–224.
- [24] K. Harada *et al.*, "Grasp planning for parallel grippers with flexibility on its grasping surface," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2011, pp. 1540–1546.
- [25] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4679–4684.
- [26] H. Zha, T. Hoshida, and T. Hasegawa, "A recursive fitting-and-splitting algorithm for 3-D object modeling using superquadrics," in *Proc. Int. Conf. Pattern Recognit.*, 1998, pp. 658–662.
- [27] S. El-Khoury, A. Sahbani, and V. Perdereau, "Learning the natural grasping component," in *Proc. IEEE/RJS Int. Conf. Intell. Robots Syst.*, 2007, pp. 2957–2962.
- [28] Z. Xue, A. Kasper, J. M. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *Proc. Int. Conf. Adv. Robot.*, 2009, pp. 1–6.
- [29] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1824–1829.
- [30] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 1628–1633.
- [31] M. Bonilla, D. Resasco, M. Gabiccini, and A. Bicchi, "Grasp planning with soft hands using bounding box object decomposition," in *Proc. Int. Conf. Intell. Robots Syst.*, 2015, pp. 518–523.



- [32] K. Nagata *et al.*, "Picking up an indicated object in a complex environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2109–2116.
- [33] N. Yamanobe and K. Nagata, "Grasp planning for everyday objects based on primitive shape representation for parallel jaw grippers," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2010, pp. 1565–1570.
- [34] N. Curtis and J. Xiao, "Efficient and effective grasping of novel objects through learning and adapting a knowledge base," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2252–2257.
- [35] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai, "Probabilistic approach for object bin picking approximated by cylinders," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3727–3732.
- [36] F. T. Pokorny, K. Hang, and D. Kragic, "Grasp moduli spaces," in *Proc. Robot., Sci. Syst.*, 2013.
- [37] T. Tsuji, S. Uto, K. Harada, R. Kurazume, T. Hasegawa, and K. Morooka, "Grasp planning for constricted parts of objects approximated with quadric surfaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2447–2453.
- [38] M. Garland, A. Willmott, and P. S. Heckbert, "Hierarchical face clustering on polygonal surfaces," in *Proc. Symp. Interactive 3D Graph.*, 2001, pp. 49–58.
- [39] K. Hang, J. A. Stork, F. T. Pokorny, and D. Kragic, "Combinatorial optimization for hierarchical contact-level grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 381–388.
- [40] K. Hang, J. A. Stork, and D. Kragic, "Hierarchical fingertip space for multi-fingered precision grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1641–1648.
- [41] 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.
- [42] Y. Zhang, A. Koschan, and M. Abidi, "Superquadrics based 3D object representation of automotive parts utilizing part decomposition," in *Proc. Int. Conf. Qual. Control Arti. Vision*, 2003, pp. 241–251.
- [43] Z. Hu, W. Wan, and K. Harada, "Designing a mechanical tool for robots with 2-finger parallel grippers," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2981–2988, Jul. 2019.
- [44] D. Sanchez, W. Wan, F. Kanehiro, and K. Harada, "Re-posing objects considering bipedal stability constraints: An approach for stability preservation during humanoid manipulation," in *Int. J. Humanoid Robot.*, vol. 17, no. 2, 2020, Art. no. 2050005.
- [45] D. Sanchez, W. Wan, and K. Harada, "Tethered tool manipulation planning with cable maneuvering," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2777–2784, 2020.
- [46] W. Wan, K. Harada, and F. Kanehiro, "Preparatory manipulation planning using automatically determined single and dual arm," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 442–453, Jan. 2020.
- [47] W. Wan, H. Igawa, K. Harada, H. Onda, K. Nagata, and N. Yamanobe, "A regrasp planning component for object reorientation," *Auton. Robots*, vol. 5, no. 43, pp. 1101–1115, 2019.
- [48] J. Ma, W. Wan, K. Harada, Q. Zhu, and H. Liu, "Regrasp planning using stable object poses supported by complex structure," *IEEE Trans. Cogn. Develop. Syst.*, vol. 11, no. 2, pp. 257–269, Jun. 2019.
- [49] W. Wan and K. Harada, "Regrasp planning using 10,000 s of grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1929–1936.
- [50] W. Wan and K. Harada, "Developing and comparing single-arm and dual-arm regrasp," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 243–250, Jan. 2016.
- [51] H. Chen, W. Wan, and K. Harada, "Combined task and motion planning for a dual-arm robot to use a suction cup tool," in *Proc. IEEE/RAS Int. Conf. Humanoids Robots*, 2019, pp. 446–452.
- [52] R. Moriyama, W. Wan, and K. Harada, "Dual-arm assembly planning considering gravitational constraints," in *Proc. IEEE/RSJ Int. Conf. Robot. Syst.*, 2019, pp. 5566–5572.
- [53] W. Wan, K. Harada, and K. Nagata, "Assembly sequence planning for motion planning," *Assem. Autom.*, vol. 38, no. 2, pp. 195–206, 2018.
- [54] A. D. Kalvin and R. H. Taylor, "Superfaces: Polygonal mesh simplification with bounded error," *IEEE Comput. Graph. Appl.*, vol. 16, no. 3, pp. 64–77, May 1996.
- [55] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Comput. Euclidean Geometry*, vol. 1, pp. 23–90, 1992.
- [56] K. Harada, T. Tsuji, S. Uto, N. Yamanobe, K. Nagata, and K. Kitagaki, "Stability of soft-finger grasp under gravity," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 883–888.
- [57] F. B. J. Mittleman, H. R. C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Visualization Comput. Graph.*, vol. 5, no. 4, pp. 349–359, Oct.–Dec. 1999.
- [58] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, vol. 96, pp. 226–231, 1996.



**Weiwei Wan** (Member, IEEE) received the Ph.D. degree in robotics from the Department of Mechano-Informatics, the University of Tokyo, Tokyo, Japan, in 2013.

He is currently an Associate Professor with the School of Engineering Science, Osaka University, Osaka, Japan. From 2013 to 2015, he was a Post-doctoral Research Fellow with the Japanese Society for the Promotion of Science, Japan, and did his Post-doc Research with Carnegie Mellon University, Pittsburgh, PA, USA. Before joining Osaka University, he was a tenure-track Research Scientist with the National Advanced Institute of Science and Technology (AIST), Tsukuba, Japan. His current research interest includes smart manufacturing and robotic manipulation.



**Kensuke Harada** (Member, IEEE) received the Ph.D. degree in engineering from the Graduate School of Mechanical Engineering, Kyoto University, Kyoto, Japan, in 1997.

He is currently a Professor with the Graduate School of Engineering Science, Osaka University, Osaka, Japan. From 1997 to 2002, he was a Research Associate with the Graduate Industrial and Systems Engineering, Hiroshima University, Hiroshima, Japan. From 2005 to 2006, he was a Visiting Scholar with the Computer Science Department, Stanford University, Palo Alto, CA. Before joining Osaka University, he was a Researcher with the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His current research interests include the mechanics and control of humanoid robots and robotic hands.



**Fumio Kanehiro** (Member, IEEE) received the Ph.D. degree in engineering from The University of Tokyo, Tokyo, Japan, in 1999.

He is currently a Group Leader with the Humanoid Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He was Special Researcher with the Japan Society for the Promotion of Science in 1998, and joined the Electrotechnical Laboratory, Agency of Industrial Science and Technology in 2000. Due to reorganization, he has been the Researcher with the Intelligent Systems Research Institute, AIST, since 2001, and a Senior Researcher in October 2007. He has been a Visiting Researcher with the LAAS-CNRS, France, since April 2007. His current research interest include the system composition and whole body motion control of the humanoid robot.