



ORIGINAL ARTICLE

Grasp planning via hand-object geometric fitting

Peng Song¹ · Zhongqi Fu¹ · Ligang Liu¹

Published online: 7 November 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Grasp planning is crucial for many robotic applications such as object manipulation and object transport. Planning stable grasps is a challenging problem. Many parameters such as object geometry, hand geometry and kinematics, hand-object contacts have to be considered, making the space of grasps too large to be exhaustively searched. This paper presents a general approach for planning grasps on 3D objects based on hand-object geometric fitting. Our key idea is to build a contact score map on a 3D object's voxelization, and apply this score map and a hand's kinematic parameters to find a set of target contacts on the object surface. Guided by these target contacts, we find grasps with a high quality measure by iteratively adjusting the hand pose and joint angles to fit the hand's instantaneous geometric shape with the object's fixed shape, during which the fitting process is speeded up by taking advantage of the discrete volumetric space. We demonstrate the effectiveness of our grasp planning approach on 3D objects of various shapes, poses, and sizes, as well as hand models with different kinematics. A comparison with two state-of-the-art approaches shows that our approach can generate grasps that are more likely to be stable, especially for objects with complex shapes.

Electronic supplementary material The online version of this article (doi:[10.1007/s00371-016-1333-x](https://doi.org/10.1007/s00371-016-1333-x)) contains supplementary material, which is available to authorized users.

✉ Peng Song
songpeng@ustc.edu.cn

Zhongqi Fu
fzq2011@mail.ustc.edu.cn

Ligang Liu
lgliu@ustc.edu.cn

¹ University of Science and Technology of China, Hefei, China

Keywords Grasp planning · 3D object · Hand model · Geometric fitting · Grasp quality

1 Introduction

Given a 3D object and a hand model, grasp planning targets at generating hand configurations (i.e., hand pose and joint angles) that can achieve stable grasps on the object. However, a hand model usually has a large number of degree of freedoms (DOFs), making the hand configuration space too large to be exhaustively searched. Many approaches have been proposed to simplify the problem by employing predefined grasps on training objects [14, 30] or primitive shapes [9, 19]. However, this kind of approaches has its own limitations. First, it limits the kinds of objects that can be grasped since some objects may not be similar to the training objects or represented well using shape primitives. Second, the generated grasping results heavily depend on the predefined grasps, restricting the diversity of output grasps. Lastly, it requires a preprocessing step to build a grasp database and/or approximate the shape of a 3D object, which could be tedious.

We observe that grasps of high quality are likely to be obtained when the instantaneous geometric shape of a hand fits the fixed shape of a rigid 3D object by adjusting the hand's configuration, see Fig. 1. Inspired by this observation, we propose to address the grasp planning problem from the perspective of geometric fitting, which is a classical technique that has been studied in many computer graphics applications, such as mesh segmentation [2], object structure recovery [28], and 3D face reconstruction [8]. Different from existing works [2, 8, 28] that fit a static 3D model (e.g., an object model) to another 3D partial model (e.g., a depth scan), this paper fits a more complex model, which is an articulated hand model with many DOFs, to a 3D object model

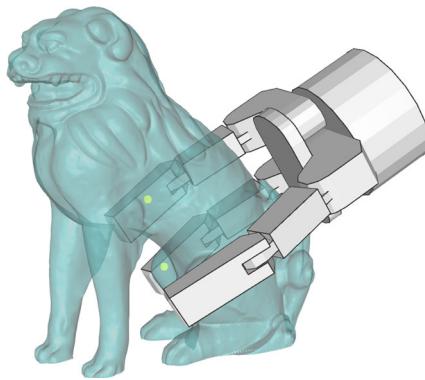


Fig. 1 Grasping a 3D object with a robot hand (i.e., the three-finger Barrett hand). Three contacts are shown as *yellow dots* (one is occluded by the finger)

to be grasped. By performing hand-object geometric fitting, the grasp planning scheme can be applied to various objects and hand models, and support generating diverse grasping modes, without being restricted by existing grasps in a grasp database.

Planning grasps using geometric fitting presents new challenges. First, a number of matched local shapes between a hand and a 3D object, i.e., hand-object contacts, need to be constructed and the set of these contacts should follow certain constraints to ensure a stable grasp. Second, the hand-object geometric fitting needs to be performed repeatedly whenever a hand updates its configuration, e.g., by changing a joint angle. This requires a fast enough computational approach to perform the fitting. Third, objects could have thin or fragile features, and these features should not be touched by the hand to avoid breaking them during the grasping.

This paper develops a general grasp planning approach based on hand-object geometric fitting. First, we build a contact score map on a 3D object's voxelization to indicate local object surfaces that are suitable to be touched by a hand. Next, we find a set of target contacts on the object surface using the contact score map and the hand's kinematic parameters such that these target contacts have high chance to be reached by the hand fingers/palm. Guided by these target contacts, we find hand grasping configurations by iteratively adjusting the hand pose and joint angles to minimize distances between the fingers/palm and the associated target contacts, during which the discrete volumetric space is employed to avoid hand-object penetration efficiently. A grasp is formed when the hand's instantaneous geometric shape fits the object's shape. To ensure a stable grasp, we evaluate the generated grasps on a quantitative quality measure and select the one with a high measure score.

Thanks to the geometric fitting scheme, our approach not only allows planning grasps on objects of various shape complexities, and hand models with a large number of DOFs, but also supports generating assorted grasping modes,

e.g., precision grasps with different numbers of fingers. We demonstrate the effectiveness of our approach on 3D objects of various shapes, poses, and sizes, as well as hand models with different kinematics, and present a comparison with state-of-the-art approaches [4, 23].

2 Related work

Existing works on grasp planning [5, 27] can be classified as empirical approaches that apply the knowledge acquired from existing grasps to synthesize new grasps, and analytical approaches that model the grasping procedure, e.g., using 3D simulation, to find desired grasps.

2.1 Empirical approaches

These approaches are commonly used for human grasping synthesis in computer animation. Aydin and Nakajima [3] animated grasping for virtual human actors and estimated a hand grasping posture for an object based on classifying the object as one of the primitive shapes and selecting an associated predefined grasp. Pollard and Zordan [22] generated real-time realistic grasping motion by combining physically based simulation with real motion-captured grasp examples. Kry and Pai [13] captured both hand motion and contact forces when grasping a real object, and transferred them onto virtual objects using physically based simulation. Li et al. [16] selected a best human grasp for a target object by shape matching between the object and hands with various fixed poses in a grasp database. Compared with [16], our geometric fitting approach allows a hand to change not only its pose but also its joint angles continuously. Thus, it can output the hand grasping configuration and the corresponding hand-object contacts, without relying on a grasp database. By this, our approach is easier and more flexible to apply to different hand models, including both robot and human hands.

More recently, Amor et al. [1] synthesized natural looking human grasping by developing a low-dimensional probabilistic grasp model in human grasp space, and training the model using motion-captured grasp examples. Ciocarlie and Allen [7] applied low-dimensional hand posture subspaces to speed up the search of grasping postures with dexterous hands. Kyota and Saito [14] developed an interactive tool for human grasping synthesis, where users can select grasp types in a grasp taxonomy, specify grasping position, and select grasping postures generated by the tool. Zhao et al. [30] synthesized human grasping motion from prerecorded motion capture data and transformed the motion into physically realistic grasping animations using a real-time physics-based motion control system.

Using a grasp database, these approaches can synthesize realistic grasping postures and/or animations. However,

building a large grasp database remains a tedious work, especially when supporting grasping with different hand models is required. Moreover, these approaches could have difficulty in synthesizing new grasps that differ greatly from the acquired grasp samples in the database.

2.2 Analytical approaches

These approaches are generally used for planning grasps in robotics. According to the order of computing hand-object contacts and a hand pose for grasping, these approaches can be classified into two classes.

The first class computes contacts on the object first, followed by finding the hand grasping configuration. Przybylski et al. [23] evaluated local symmetry properties of a 3D object represented by its medial axis to find desirable contacts. Rosales et al. [26] addressed the problem of configuring a robot hand so as to grasp a 3D object while satisfying a number of specified hand-object contacts. Ye and Liu [29] synthesized human grasping motions by finding contact points on the object surface using physics-guided sampling and reconstructing the grasping motion using inverse kinematics. Li et al. [17] identified promising grasping spots on a 3D object surface by wrapping multiple cords around the object, and computed grasping configurations from these spots simply by closing the hand until its fingers/palm contact the object surface.

The second class computes the hand grasping pose first, followed by the contact points that can be reached from the pose. Miller et al. [19] generated hand grasping positions and directions for a given object using predefined grasps on shape primitives, and computed the contact points by closing the hand until contacting. Goldfeder et al. [9] represented a 3D object using a superquadric decomposition tree, and

applied it to obtain good grasp candidates, which are further sampled and evaluated in a grasping simulator [18] to find stable grasps. Huebner et al. [11] approximated a 3D object with primitive boxes to identify parts of the object that can be grasped, and found grasping configurations using a close-until-contact procedure.

Compared with the above works, we generate grasps effectively by performing hand-object geometric fitting in the discrete volumetric space. This brings several advantages. First, our approach can be more flexibly applied to general 3D objects and different hand models, as well as support a wide variety of grasping modes. Second, by incorporating exact object geometry, as well as geometry and full range of kinematic parameters of the hand in the fitting process, our generated grasps are likely to be stable.

3 Overview

Our input is a 3D object and a hand model, both represented by triangular meshes. We assume that the object and the hand are solid to avoid considering shape deformation during the grasping. After users specify a grasp style (e.g., a precision or power grasp) and the fingers/palm that should touch the object, our goal is to generate a set of high-quality grasps that satisfy the specified requirements, among which users can select a desirable one to perform a given task, e.g., robotic manipulation (Fig. 2).

3.1 Hand models

We model two robot hands and one human hand for our grasping experiments, see Fig. 3. Each hand is an articulated model, and movements of the fingers are controlled

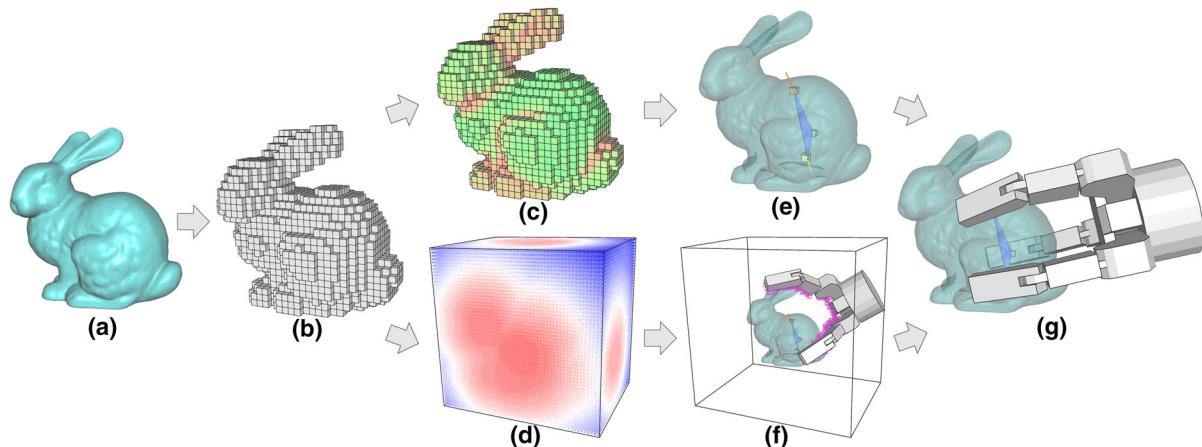


Fig. 2 Overview: **a** a BUNNY model, **b** its voxelization, **c** contact score map (low and high scores are colored in red and green, respectively), **d** distance transform map (small and large distances are colored in red and blue, respectively), **e** select a set of three target contacts from the

contact score map, **f** adjust a Barrett hand's configuration to fit its shape with that of the target contacts (voxels occupied by the fingers/palm are colored in purple), and **g** an output grasp

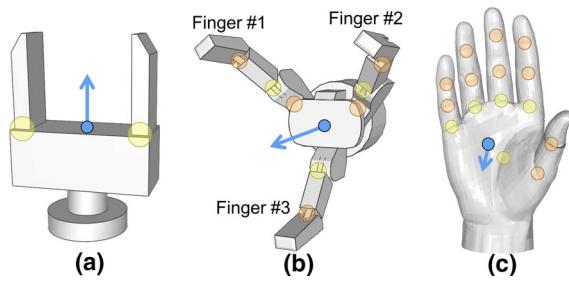


Fig. 3 Our hand models: **a** parallel gripper, **b** Barrett hand, and **c** human hand. Hand joints are highlighted as *yellow dots* (proximal joints) and *orange dots* (other joints) while palm centers and directions are colored in *blue*

by the joint angles. The configuration of each hand is completely defined by a combination of intrinsic DOFs (finger joint angles) and extrinsic DOFs (hand position and orientation).

- (1) **Parallel gripper** The first one is a two-finger gripper with one intrinsic DOF, see Fig. 3a. Each finger of the gripper is controlled to do translational motion symmetrically to grip an object. Thus, the gripper usually needs to be centered over an object for both fingers to come in contact with the object.
- (2) **Barrett hand** The second one is a three-finger Barrett hand, see Fig. 3b. Finger #1 and #2 can spread synchronously up to 180° about the palm, while finger #3 is stationary with the palm. The Barrett hand has only four intrinsic DOFs: one for the spread angle of finger #1 and #2, and three for the joint angles of each finger's proximal joint (the yellow dots in Fig. 3b). More details about this hand can be found in [18].
- (3) **Human hand** The last one is a five-finger human hand, where each finger has three joints, see Fig. 3c. For each finger, we describe the movement of distal joints and proximal joints using one DOF and two DOFs, respectively. Hence, this hand model has 20 intrinsic DOFs: four for each finger.

3.2 Overview of our approach

Geometric fitting of two models represented by triangular meshes is challenging, especially when one model (i.e., the hand) has a number of DOFs. We address the problem in an efficient manner by taking advantage of the object's voxelization. Figure 2 outlines our major steps:

- First, we construct a large bounding box of the object, and employ it as an interaction volume for the hand-object geometric fitting. We voxelize the bounding box and identify each voxel as empty, partial, or full if the voxel is outside, intersected with, or inside the object sur-

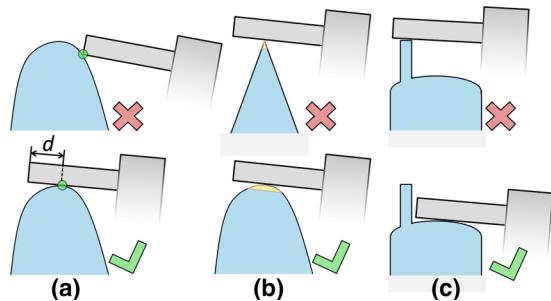


Fig. 4 Measures on a single contact: **a** distance d between a contact (green point) and a finger's boundary, **b** estimated area of the contact region (colored in yellow), **c** whether the contact falls onto weak object features

face [20] (Fig. 2b). We further compute a contact score map on the voxelization to identify which local object surfaces (within partial voxels) are suitable to be touched by the hand fingers/palm for grasping (Fig. 2c).

- Second, we select a set of target contacts on the object surface based on the contact score map, and the hand's kinematic parameters (Fig. 2e). Our criteria on selecting the target contacts are: (1) the ideal grasp with these contacts should have a high score on a quantitative quality measure that we formulate; (2) the contacts should have a high chance to be reached by the fingers/palm. By this, the set of target contacts becomes an effective guidance on posing the fingers/palm to form high-quality grasps.
- Third, after associating the target contacts to the fingers/palm that were specified by users, we iteratively adjust the hand pose and joint angles such that the hand's shape can fit better to the target contacts on the object surface (Fig. 2f). During the geometric fitting, we avoid hand-object penetration efficiently by employing a signed distance transform map defined on the object's voxelization (Fig. 2d). Once the fingers/palm touch the object surface, a grasp is formed and we can obtain the corresponding hand configuration and estimate actual hand-object contacts (Fig. 2g).

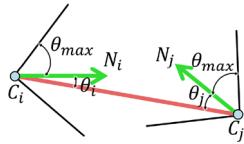
4 A quantitative grasp quality measure

Given an object and a hand, there could exist a number of grasps that satisfy the requirements of a given task. To select optimal ones, various grasp quality measures [12, 24, 25] have been proposed to quantify the goodness of a grasp. In this paper, we formulate a quantitative grasp quality measure that can be easily computed from the corresponding hand-object contacts.

In particular, our quality measure considers the contacts in three different levels, i.e., single contact level, contact pair level, and whole contact set level:

(1) *Sustainability of each single contact* Each single contact should be sustainable when a finger exerts forces. Thus, we need to avoid contacting an object surface with a finger's corner to prevent the finger from slipping easily. We evaluate this measure by computing a distance d between the contact point and the finger's boundary (Fig. 4a). Second, fingers need to avoid contacting object features that are sharp, which makes the fingers difficult to pose as desired. We evaluate this measure by estimating the contact region's area A (Fig. 4b). Third, fingers also need to avoid contacting object features that are weak (Fig. 4c). This requires an offline shape analysis on the object to identify weak shape features [31]. We define $S = 0$ if the contact falls on weak features and $S = 1$ otherwise. By summarizing the three contact measures for all hand-object contacts, we obtain: $M_1 = (\sum_{i=1}^n (d_i + \alpha A_i) S_i)/n$, where n is the number of contacts, and α is a weight (we set $\alpha = 2$ in our experiments).

(2) *Stability of each pair of contacts* The cone of friction is a geometric interpretation of the maximally allowed angle θ_{\max} between the surface normal and the applied force vector at a contact point.



Specifically, for each pair of contacts from different fingers, we use them to define a connecting line. If this line lies within both cones of friction at the contact points, we regard that the pair of contacts is stable [1]. For each stable contact pair, we evaluate it based on an averaging angle between the connecting line and the surface normal at each contact, i.e., $\theta = (\theta_i + \theta_j)/2$, where a small θ results in a more stable contact pair. When there exist $m > 0$ stable contact pairs, we evaluate them as: $M_2 = \sum_{l=1}^m (1 - \theta_l/\theta_{\max})$. When there is $m = 0$ stable contact pair, we set $M_2 = 0$.

(3) *Centroid and regularity of the contact polygon/polyhedron* A contact polygon/polyhedron can be computed as the convex hull of all contact points. Intuitively, a shorter distance D between the object's center of mass and the centroid of the contact polygon/polyhedron results in less effect of inertial and gravitational forces on the grasp, making the grasp more stable. Grasp stability can be further improved by distributing the contact points in an uniform way on the object surface [21], which can be measured by regularity R^1 of the contact polygon/polyhedron [6]. By combining these two measures, we obtain: $M_3 = R - \beta D$, where β is a weight

¹ Regularity indicates if a polygon/polyhedron has all the same qualitative angles and all the edges of similar length (regular), or not (irregular). We normalize the regularity value within range (0,1].

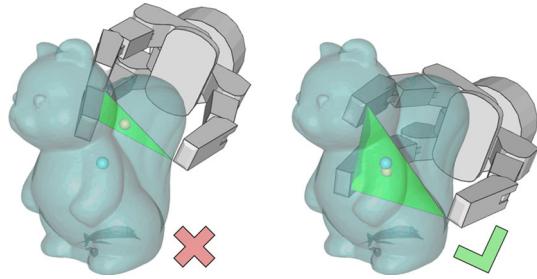


Fig. 5 Contact polygons (i.e., triangles) of two grasps on the SQUIRREL. Left an irregular contact polygon whose centroid (in yellow) is far from that (in blue) of the object; right a regular contact polygon results in a more desirable grasp

and we set $\beta = 0.3$ in our experiments. Figure 5 shows two grasp examples evaluated based on their contact polygons.

By summarizing the above three measures, we formulate a quantitative quality measure on a given grasp: $M = M_1^{w_1} M_2^{w_2} M_3^{w_3}$, where we set the weights $w_1 = 1$, $w_2 = 1$, and $w_3 = 0.5$ in all our experiments. By employing the multiplicative combination, we can identify usable grasps by checking if $M > 0$. For example, $M = 0$ could indicate that an unusable grasp caused by $M_2 = 0$ when there exists no stable pair of contact forces applied on the object. Moreover, our grasp measure can identify unstable grasps of various cases such as the ones shown in Fig. 6.

5 Computing target hand-object contacts

One efficient way to generate high-quality grasps is finding a set of target contacts on an object surface to guide the search. Due to the continuity of the object surface, there exist numerous points on the surface that can be potential contact points, some of which are very close to one another in the physical space. Nevertheless, the dimension of fingers requires a certain distance between the contact points such that the fingers can be posed on the object surface without colliding with one another during the grasping.

We sample points on the object surface using its voxelization to reduce the number of candidate contacts and get rid of contacts that are too close to one another. Moreover, we build a contact score map on the object's voxelization to indicate which local object surfaces are suitable to be contacted by a hand. We further apply the score map together with the hand's kinematic parameters to select a set of target contacts, aiming at generating grasps with a high-quality measure described in Sect. 4.

5.1 Build a contact score map

A smooth object region usually results in a stable hand-object contact since human/robot fingers mostly have flat (or close to

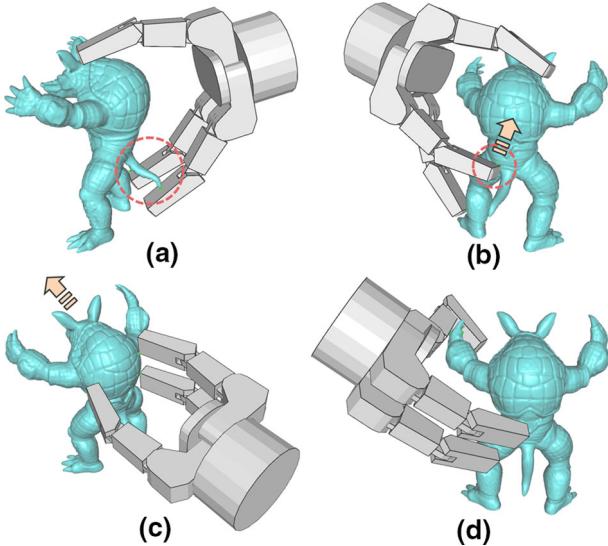


Fig. 6 Example of unstable grasps that can be identified by our quality measure: **a** a contact (in red circle) that could break the fragile tail of the ARMADILLO, **b** an unsustainable contact (in red circle) that locates at a finger's corner, **c** the object could escape from the hand during the grasping, and **d** the object could have undesired movements during the grasping due to its irregular contact polyhedron

flat) surfaces. Thus, we try to avoid picking sharp, concave, or uneven object surface regions as target contacts. Inspired by this observation, we build a contact score map on an object's voxelization by computing a score value for the local object surface within each partial voxel to measure suitability that a finger touches the local surface.

The score value of each partial voxel V_i is computed using the centroid and normal of the local object surface within V_i , as well as those of V_i 's neighboring voxels (we use a $5 \times 5 \times 5$ neighborhood in our experiments). We find the local object surface within V_i , denoted as S_i , by clipping the object mesh triangles intersected with V_i using the six planes of the voxel. We estimate the centroid of S_i by averaging the centroid of every triangle in S_i and projecting the point back onto the object surface. The normal of S_i is estimated simply by averaging the normal of every triangle in S_i . In the following, we will not differentiate between the centroid and normal of S_i and those of V_i . Figure 7a shows estimated centroids and normals for the voxelization of a SQUIRREL model.

For a partial voxel V_i with centroid p and normal \mathbf{n} , we compute a score for V_i with the following steps. First, we compute a distance between the centroid of each neighboring partial voxel and the oriented plane defined by p and \mathbf{n} , see Fig. 8 (top). By averaging all computed distance values, we obtain a score component C_d for V_i . Voxels with large C_d should be avoided since they indicate sharp or large concave object regions, see Fig. 8a, b. Second, we compute an angle between the normal of each neighboring voxel and \mathbf{n} , see Fig. 8 (bottom). By averaging all computed angles,

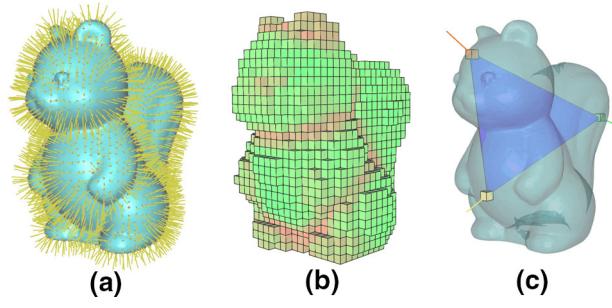


Fig. 7 **a** Centroids and normals of partial voxels. **b** Contact score map on the voxelization (low and high scores are colored in red and green, respectively). **c** Three target contact voxels and the corresponding contact triangle, where the seed voxel is in green, and the other two are in yellow and orange, respectively

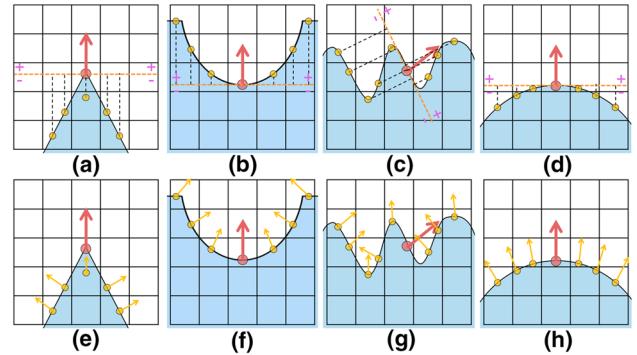


Fig. 8 Compute distance components for **a** sharp, **b** concave, **c** uneven, and **d** smooth object regions. Compute angle components for **e** sharp, **f** concave, **g** uneven, and **h** smooth regions. Centroids and normals of the target voxel and neighboring voxels are in red and yellow, respectively

we obtain the other score component C_a for V_i . Voxels with large C_a should be avoided since they indicate highly curved regions, see Fig. 8 f, g. Lastly, we calculate a score for V_i by combining the normalized distance and angle components using: $1/(C_d/L + C_a/\gamma)$, where L is the size of the $5 \times 5 \times 5$ neighborhood, and $\gamma = 180$ (i.e., maximally allowed angle between two 3D vectors). We build a contact score map by computing a score for every partial voxel, and further apply a mean filter to smooth the map, see Fig. 7b for an example.

5.2 Find a set of target contacts

We develop a randomized greedy approach to select a set of N target contacts, where N is specified by the user and $N \geq 2$. Without loss of generality, we describe our greedy approach by taking selecting $N = 3$ target contacts as an example (see Fig. 7c).

We select the target contact voxels one by one, guided by the grasp quality measure described in Sect. 4. Here, we list the requirements for selecting each contact voxel. First, we select a seed contact voxel V_1 (i.e., the green one in Fig. 7c)

whose score value is larger than a threshold S_{thres} (we set $S_{\text{thres}} = 0.6$ in our experiments). Next, we select one more contact voxel V_2 that satisfies three requirements: (1) score value of V_2 is larger than S_{thres} ; (2) length of the vector $\overrightarrow{V_1 V_2}$ is within a certain range (determined by the minimally and maximally allowed distances between fingertips of a hand model) such that both contact voxels can be reached by the fingers; (3) angle between the normal of V_1 and $\overrightarrow{V_2 V_1}$ is less than θ_{\max} (determined by the object and hand material properties, we set $\theta_{\max} = 45^\circ$ in our experiments) to satisfy the friction cone requirement; the same applies to the angle between the normal of V_2 and $\overrightarrow{V_1 V_2}$. Besides the above three requirements, we select the third contact voxel V_3 with two more requirements that are enforced on the contact triangle: (1) lengths of $\overrightarrow{V_3 V_1}$ and $\overrightarrow{V_3 V_2}$ are close to $\overrightarrow{V_1 V_2}$ to satisfy the regularity requirement; (2) centroid of the contact triangle $V_1 V_2 V_3$ is close to the object's center of mass.

In our implementation, for each target contact voxel, we find a number of candidates that satisfy the above requirements and then randomly pick one from the candidates. In the case that we cannot find any candidate for a certain contact voxel (e.g., V_3), we pick a different seed voxel and repeat the search process. Thanks to the limited number of partial voxels (e.g., 2478 partial voxels in Fig. 7b), it usually takes a short time (i.e., less than a second) to find a set of N target contact voxels successfully.

6 Generating high-quality grasps

Given a set of target contact voxels, we extract their centroids and normals to represent target hand-object contacts. Next, we associate each target contact with a finger/palm, see Fig. 9a. Given the target contacts and the associations, we employ forward kinematics rather than inverse kinematics to find hand grasping configurations due to its simplicity for hand models with a large number of DOFs. Our grasp planning approach can be applied to different hand models and different numbers of target contacts. In this paper, we take the Barrett hand and a set of three target contacts as an example to illustrate our approach, see Fig. 9.

To generate a grasp, we first set an initial hand configuration based on the set of target contacts. Next, we adjust hand pose and joint angles such that the hand's shape can fit the object's shape, and the target contacts can be reached by the associated fingers/palm, during which we avoid hand-object penetration efficiently using the object's distance transform map. A grasp is generated when the fingers/palm touch the object surface, forming actual contacts with the object. We develop a clipping-based approach to estimate these contacts, and employ them to evaluate the grasp. Our approach has five key components, detailed as follows.

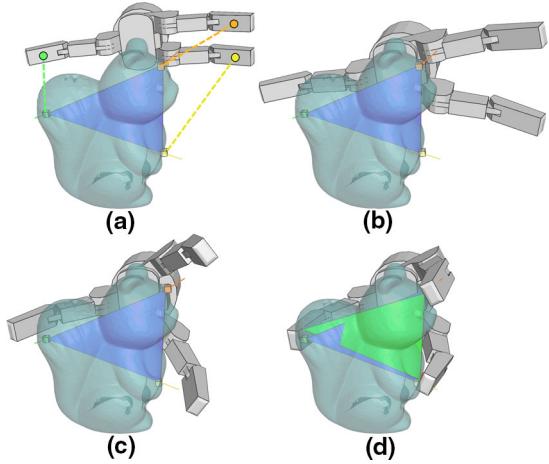


Fig. 9 **a** A set of three target contact voxels (associated fingers are highlighted with dashed lines). **b** Set initial hand configuration. **c** Adjust hand pose and joint angles for grasping. **d** A generated grasp, whose contact triangle (in green) is similar to the target one (in blue)

6.1 Set an initial hand configuration

We first open the fingers as large as possible such that the hand can grasp a larger object. For a given target contact triangle (i.e., $N = 3$), we set the hand orientation such that its palm direction \mathbf{n}_{palm} (see again Fig. 3b) is consistent with the triangle's normal. Next, we set hand position such that the hand palm center is located at a fixed distance from the triangle center along \mathbf{n}_{palm} . After that, we rotate the hand along \mathbf{n}_{palm} such that the thumb (e.g., finger #3 of the Barrett hand) is aligned with the seed contact. We also rotate the other two fingers around the palm such that each of them is aligned with its associated target contact, see Fig. 9b.

6.2 Adjust the hand configuration for grasping

Starting from the initial configuration, we employ forward kinematics to find a hand grasping configuration. In detail, we keep adjusting the hand pose and joint angles to reach the target contacts by iteratively performing the follows (Fig. 9c): (1) close the three fingers by changing the proximal joint angles; (2) translate the hand such that the centroid of the triangle formed by three fingertips is consistent with the centroid of the target contact triangle; (3) rotate the hand along \mathbf{n}_{palm} such that the thumb is aligned with the seed contact; (4) rotate each of the other two fingers around the palm to make it aligned with its associated target contact; (5) make the reversed normal of each fingertip's surface consistent with the normal of the associated target contact by changing the corresponding distal joint angle. A grasp is formed when all fingers/palm that are associated with a target contact stop their movements by touching the object surface (Fig. 9d).

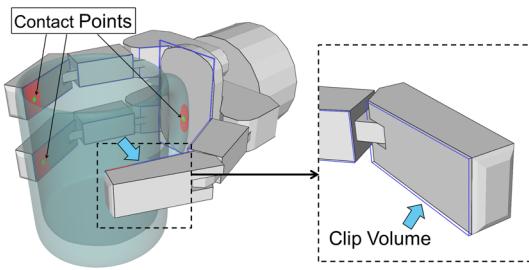


Fig. 10 Estimate hand-object contacts. *Left* four contacts (one is occluded) when grasping a MUG model with the Barrett hand. Clip volume, clipped triangles, and contact point related to each finger/palm are colored in blue, red, and green, respectively; *right* a zoom-in view of a finger's clip volume

6.3 Avoid hand-object penetration

During the adjustment of the hand configuration, the fingers/palm could penetrate the object. Thus, we need to avoid hand-object penetration by estimating the distance between each finger/palm and the object surface. To achieve this, we keep identifying voxels occupied by a finger/palm (see the purple voxels in Fig. 2f) and finding the smallest distance d_{\min} (unit: number of voxels) among the voxels by looking up the signed distance transform map.² If d_{\min} is larger (smaller) than zero, it means that there exists no penetration (penetration) between the finger/palm and the object.

Based on d_{\min} , we can further speed up the hand adjustment in the previous component: (1) when d_{\min} is large (e.g., ≥ 2), we adjust the hand configuration with a larger step, e.g., larger changes on the joint angles; (2) otherwise, we slow down the adjustment to avoid unintentional hand-object penetration.

6.4 Estimate hand-object contacts

When a finger/palm's $d_{\min} = 0$, there exist voxels that are occupied by both the finger/palm and the object surface, in which the finger/palm has a very high chance to contact the object surface. In such case, we need to identify if the finger/palm touches the object, and further estimate the contact (including contact point, direction, and size) if it exists.

We estimate contacts between a hand and an object by attaching a clip volume for each finger/palm, see Fig. 10 (right). Such volume is constructed by offsetting the surface of a finger/palm along its normal direction, which is a box-like shape for the Barrett hand and a quasi-regular shape for the human hand. When a finger/palm moves close

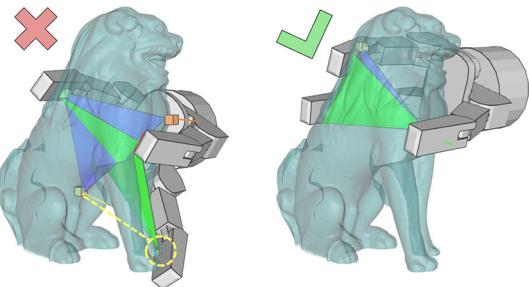


Fig. 11 *Left* an undesired grasp, in which an actual contact (yellow circle) is far from the target one. *Right* a high-quality grasp, in which the actual contact triangle (in green) is very similar to the target one (in blue)

to the object surface, we identify triangles of the object model that locate within the voxels occupied by the finger/palm, among which we further identify a subset of triangles that are intersected with the clip volume attached with the finger/palm. We consider that there is no contact if we cannot find any intersected triangle. Otherwise, we clip the intersected triangles using the clip volume to estimate the contact. In detail, the contact point and direction are estimated by averaging the centroids and normals of all clipped triangles, respectively, while the contact size is estimated by accumulating the area of all clipped triangles, see Fig. 10 (left).

6.5 Validate generated grasps

For a generated grasp, we estimate all actual hand-object contacts to evaluate it on the quality measure described in Sect. 4. Although these actual contacts could deviate a bit from the target ones due to kinematic constraints of the hand model [Figs. 9d, 11 (right)], the grasp quality is usually good since we have ensured high quality for the target contacts in Sect. 5. When the actual contacts deviate much from the target ones, e.g., finger movements blocked by object features [Fig. 11 (left)], quality of the actual grasp could be much lower than that of the respected one. For such case, we select a new set of target contact voxels and apply it to generate another grasp result.

Our grasp planning procedure terminates when we find a high-quality grasp, or the maximum number of iterations is reached, indicating that the specified hand model is not suitable for grasping the target object.

7 Results

This section presents various experimental results to demonstrate the effectiveness of our approach.

² In the signed distance transform map, empty, partial, and full voxels are assigned positive, zero, and negative distance values, respectively.

7.1 Parameters

After selecting a hand model, users need to specify the grasp style, as well as which fingers/palm should touch the object for grasping by mouse clicking on the corresponding fingers/palm. When a precision grasp style is selected, our system associates the fingertips of the selected fingers to the target contacts. Thus, the maximum number of target contacts in a precision grasp is the same as the number of fingers (i.e., 2, 3, and 5 for the gripper, Barrett, and human hands, respectively). When a power grasp style is selected, not only the fingertips but also the palm center (if the palm is allowed to touch the object) are associated with the target contacts. Moreover, when planning power grasps, our system encourages the proximal links of the fingers to touch the object for making the grasps more stable. Thus, the number of actual contacts in power grasps could be larger than the number of target ones.

Besides the user specified parameters, there are some parameters related to the geometric fitting. First, the size of an

object's bounding box (see Fig. 2d, f) is determined by the object and the hand dimensions such that the hand can be enclosed in the box when performing hand-object geometric fitting. Second, the voxelization resolution $W \times H \times D$ of an object (see Figs. 2c, 7b) is set in a way such that $\max(W, H, D) = 30$ while the other two dimensions are calculated proportionally, as a tradeoff between accuracy to represent object shape features and computational complexity to find target contacts.

7.2 Performance

We implemented our grasp planning approach in C++ and OpenGL on a desktop computer (3.4 GHz CPU and 8 GB memory). Thanks to the speeded-up geometric fitting using an object's voxelization (e.g., target contact voxels), our approach can generate grasps fairly efficiently. Figure 12 presents the timing statistics for the generated grasps, most of which can be completed within a few seconds. The timing performance depends on several factors: shape complexity of

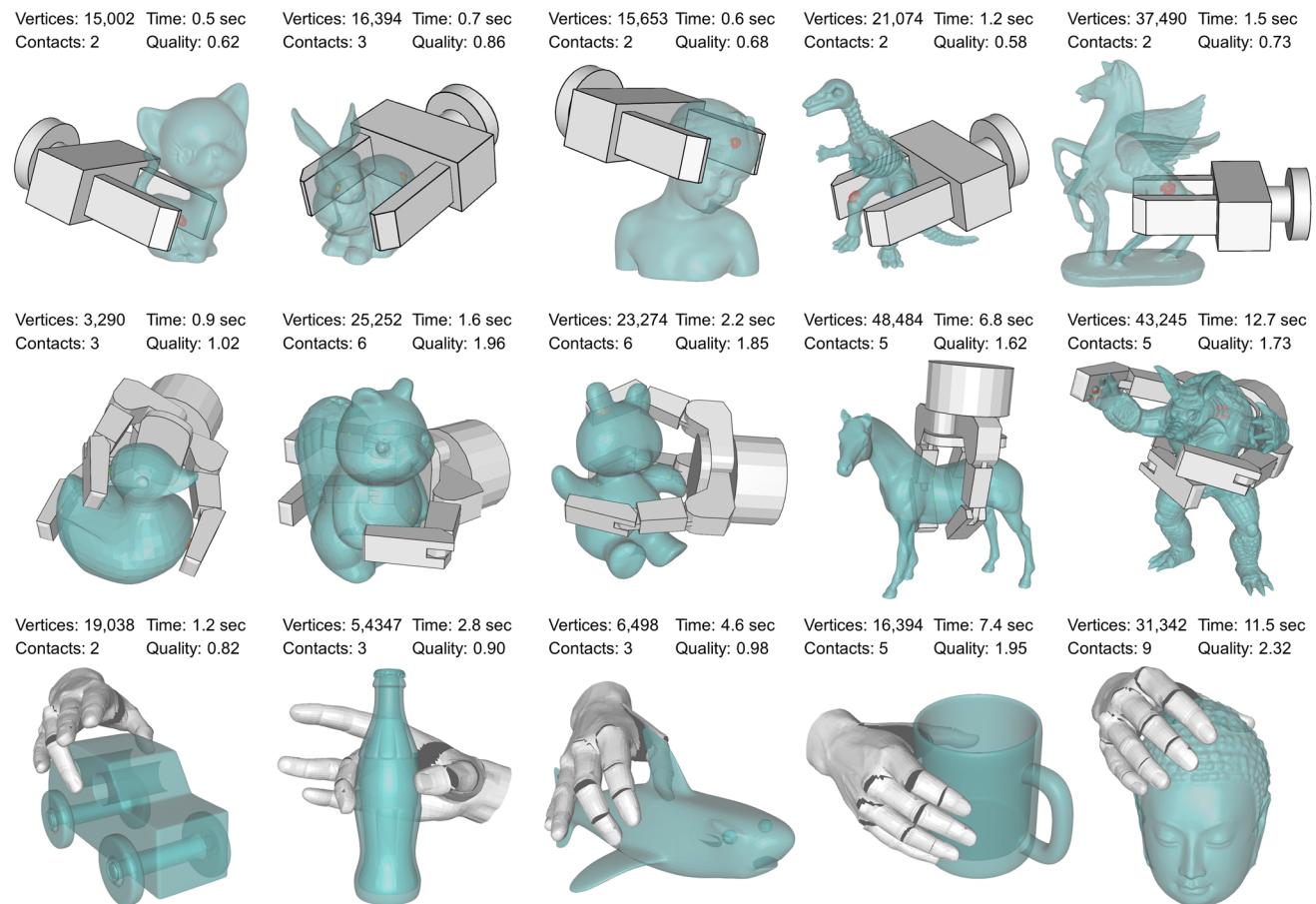


Fig. 12 Our generated grasps on various objects with *top* the gripper, *middle* Barrett hand, and *bottom* human hand. From *left* to *right* and then *top* to *bottom* KITTEN, BUNNY, BIMBA, DINOSAUR, PEGASO, DUCK, SQUIRREL, TEDDY, HORSE, ARMADILLO, TOY CAR, BOTTLE,

SHARK, MUG, and BUDDHA HEAD. The number of model vertices, time to generate the grasp, number of contacts, and grasp quality score are shown on *top* of each grasp result

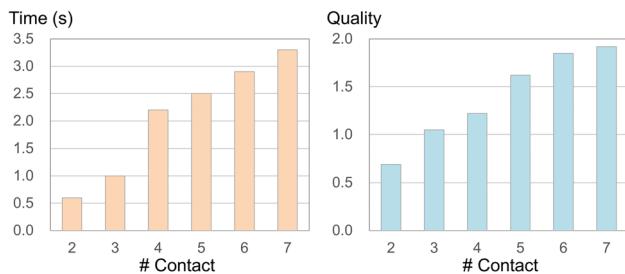


Fig. 13 *Left* computation time versus number of actual contacts for the Barrett hand. *Right* grasp quality versus number of actual contacts for the Barrett hand

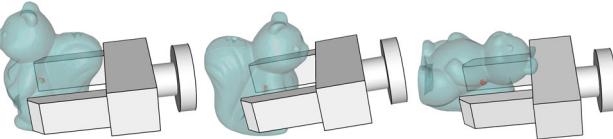


Fig. 14 Grasp a SQUIRREL with three different poses

the object, resolution of the object mesh model, shape complexity and number of DOFs of the hand. Figure 13 shows the computation time and grasp quality measure with relative to the number of actual contacts denoted as N' for the Barrett hand, respectively. Note that the computation time increases significantly from $N' = 3$ to $N' \geq 4$ since searching a target contact triangle is much cheaper than searching a target contact polyhedron. The grasp quality improves also when N' increases since a larger N' usually results in a larger number of stable contact pairs, making M_2 larger (see again Sect. 4).

7.3 Grasp results

Our approach allows generating high-quality grasps on objects with various poses, sizes, and shapes, as well as different hand models with various grasping modes.

Different object poses Our approach allows planning grasps on an object with different poses, see Fig. 14 for an example. Intuitively, when the pose of an object is fixed, our approach allows a hand to grasp the object from different approaching directions, which can be specified by users according to a certain task.

Different object sizes Our approach allows planning grasps on objects of different sizes. Figure 15 shows the grasping results on the BUNNY with three different sizes, i.e., 9, 15, and 30 cm width in our grasping simulation system. Note that for the smallest BUNNY model, we grasp it using a two-finger pinch rather than a three-finger grip to avoid touching the small (and weak) ears of the model.

Our approach takes non-zero genus shape into account implicitly by considering hand kinematic parameters when

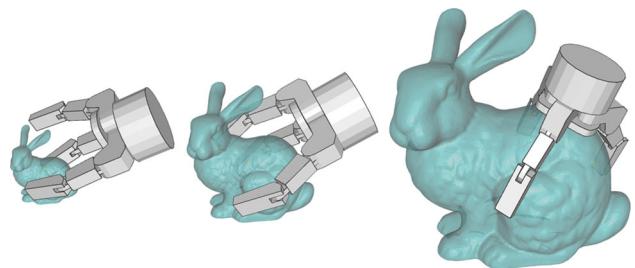


Fig. 15 Grasp BUNNY of three different sizes using a two-finger pinch, a three-finger grip, and a power grasp

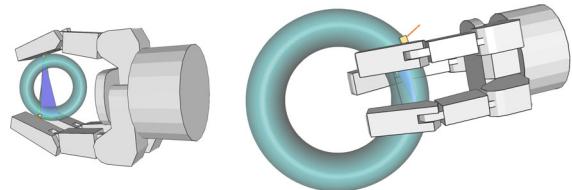


Fig. 16 Grasp TORUS of *left* a small size (7 cm width) and *right* a large size (16 cm width)

selecting a set of target contacts such that we can automatically find object components that are suitable to be grasped by the hand, even when the object size changes. Figure 16 (left) shows that the fingers need to touch outer boundary of the smaller TORUS to grasp it while Fig. 16 (right) shows that the fingers can go through the hole of the larger TORUS when grasping.

Different object shapes Figure 12 shows that our approach allows grasping a wide variety of objects with different shape complexity and topology. Thanks to the contact score map, our approach is able to avoid grasping unsuitable object features, such as thin legs of the HORSE and sharp fins of the SHARK.

Different hand models Our approach supports planning grasps with both robot and human hands. Figure 12 demonstrates our grasping results on the three hand models described in Sect. 3.

Different grasp modes Our approach supports various grasping modes. Figure 12 shows a precision grasp with three contacts on the DUCK, and a power grasp with six contacts on the TEDDY. Figure 17 shows grasping a BALL with the human hand using four different modes, i.e., two precision grasps and two power grasps.

7.4 Quantitative comparison

We compare our approach with two state-of-the-art grasp planning approaches: (1) a geometric sampling-based approach (GS) [4] that employs sampled points and normals

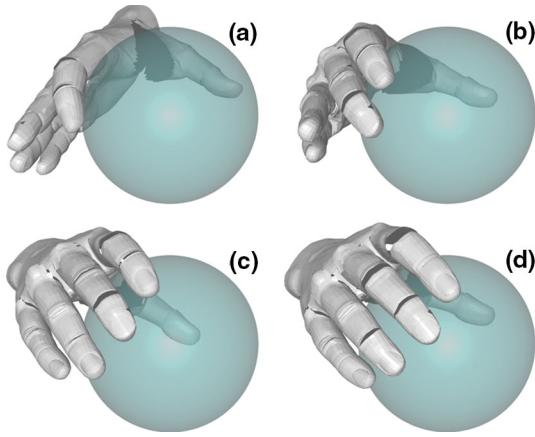


Fig. 17 Grasp a BALL with: **a** a two-finger pinch, **b** a three-finger grip, **c** four- and **d** five-finger grasps

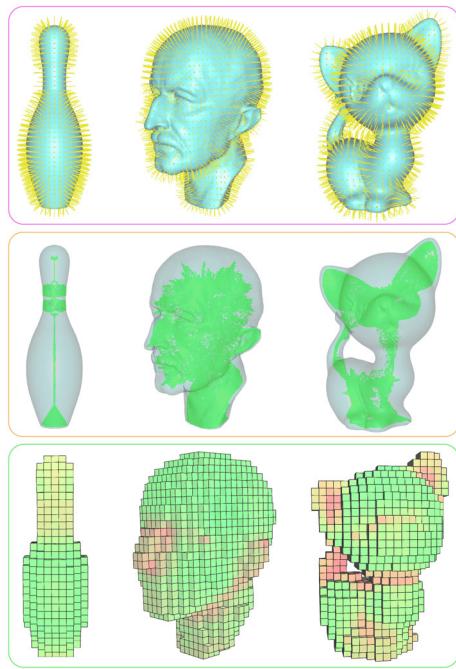


Fig. 19 Intermediate models for planning grasps created by GS, MA, and our approach (from *top* to *bottom*). From *left* to *right*: BOWLING, MAX PLANCK, and KITTEN

on the object surface to guide posing a hand for grasping; (2) a medial axis-based approach (MA) [23] that extracts local symmetry information from the medial axis of a 3D object model to guide the search of good grasps. To facilitate comparing grasp quality, we evaluate the grasps generated by GS, MA and ours using the same quality measure described in Sect. 4 (Fig. 18).

We perform the comparison on three object models, i.e., BOWLING, MAX PLANCK, and KITTEN, with the Barrett hand. To give an intuition of grasps to be generated by the three approaches, Fig. 19 shows the sampled geometry of GS, medial axis of MA, and our contact score map. Apparently, our contact score map encodes the most rich geometric information for grasping such as the sharp nose in MAX PLANCK and highly curved regions in KITTEN that should not be touched by the hand, followed by MA that encodes local symmetric structures.

We employ the three approaches to generate 100 usable grasps (i.e., quality score larger than 0). Table 1 shows average computation time and quality for the generated grasps, from which we can see that the speed of our approach is comparable to that of GS and MA, especially for objects with complex shape. More importantly, our generated grasps have a higher chance to be stable. This is because we find a set of desirable target contacts to guide the search of hand grasp configuration rather than randomly selecting target contacts as in GS or simply using the local

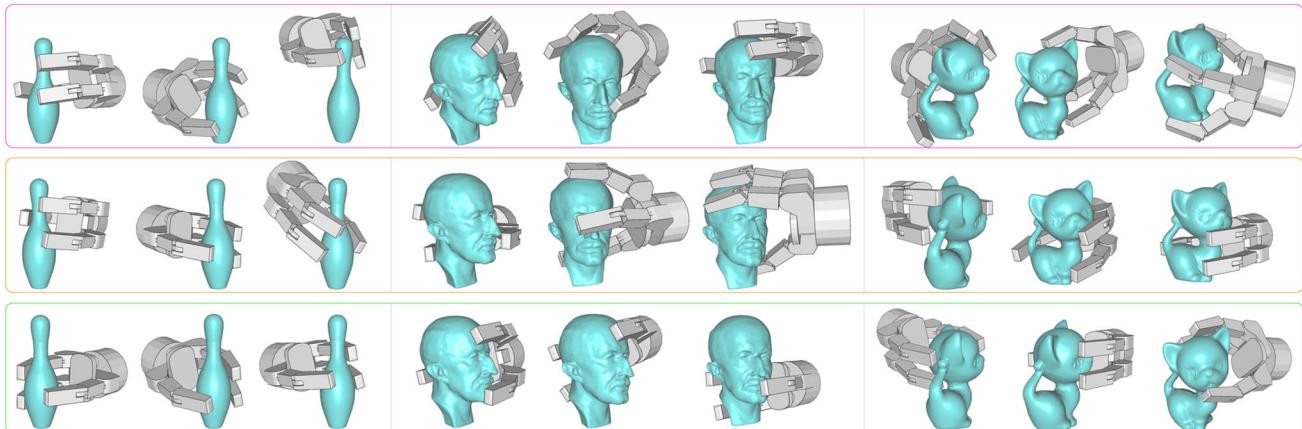


Fig. 18 Three typical grasps on BOWLING, MAX PLANCK, and KITTEN (from *left* to *right*) generated by GS, MA, and our approach (from *top* to *bottom*)

Table 1 Average computation time and quality score for 100 usable grasps generated by GS, MA, and our approach

	GS		MA		Ours	
	Avg time (s)	Avg quality	Avg time (s)	Avg quality	Avg time (s)	Avg quality
Bowling	0.5	0.96	0.3	1.18	0.6	1.36
Max Planck	2.6	0.61	1.7	0.85	1.3	1.22
Kitten	3.7	0.52	1.2	0.92	1.8	1.06

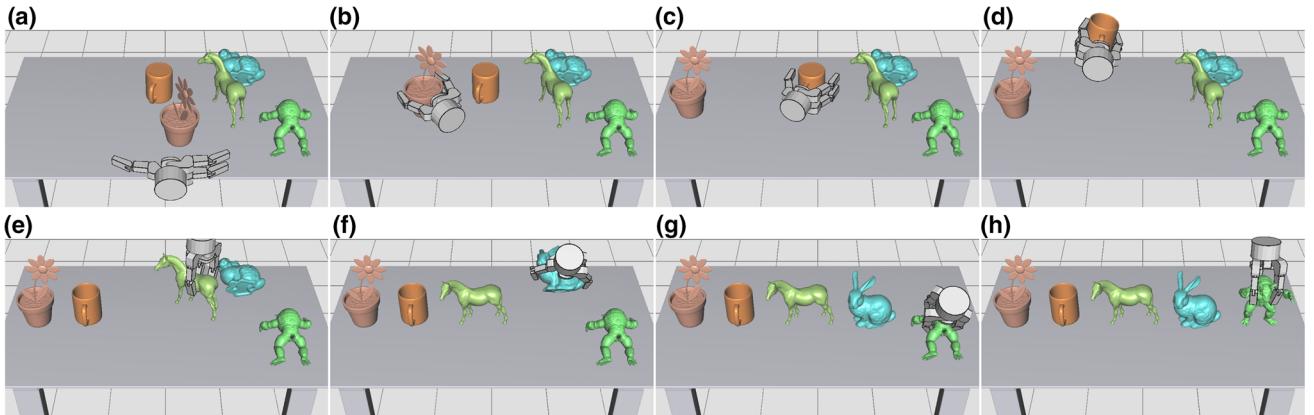


Fig. 20 Rearrange five virtual objects on a table by grasping and manipulating them with the Barrett hand

symmetric structures as in MA. This is also validated by typical grasps generated by the three approaches as shown in Fig. 18.

7.5 Organizing objects on a table

We demonstrate the power of our approach by applying it for an application of organizing virtual objects on a table. Initially, there are five objects that are disorderly arranged on the table (Fig. 20a). Our goal is to organize these objects and put them to their target position and orientation as shown in Fig. 20h. For each object, we generate grasps with the Barrett hand according to the hand approaching direction, i.e., from the hand's initial pose to the object's initial pose. Given the hand's initial and grasping configurations, we animate the procedure of grasping an object using interpolation.

In particular, we rearrange the objects on the table with the following steps (see accompanying video for the animation). First, we grasp the FLOWER POT and put it on the left of the table (Fig. 20b). Next, we grasp the MUG and rotate it in the mid-air to put its rim upward (Fig. 20c, d). After putting down the MUG, we grasp and reposition the HORSE and BUNNY (Fig. 20e, f). Lastly, we grasp and lift the ARMADILLO up and further rotate it to make it face the virtual camera (Fig. 20g, h).

8 Conclusion

We have presented a grasp planning approach based on hand-object geometric fitting. Given the grasp style and fingers/palm that should touch an object specified by users, our approach can automatically find grasps that satisfy the requirements within a few seconds. By considering the object shape, hand kinematics, and hand-object contacts during the fitting process, our generated grasps are of high quality and likely to be stable.

Compared with previous methods that employ predefined grasps, this work provides a novel and general computational solution for grasp planning with a number of novel elements: (1) formulate a quantitative grasp quality measure based on hand-object contacts; (2) select a set of target contacts based on a contact score map and a hand's kinematic parameters; (3) speed up hand-object geometric fitting by taking advantage of the discrete volumetric space; and (4) estimate actual hand-object contacts by attaching a clip volume with each finger/palm. Powered by these novel technical components, we can offer an effective solution for planning grasps on a wide variety of objects and hand models. Achievements of our approach are evidenced by various experimental results presented in this paper.

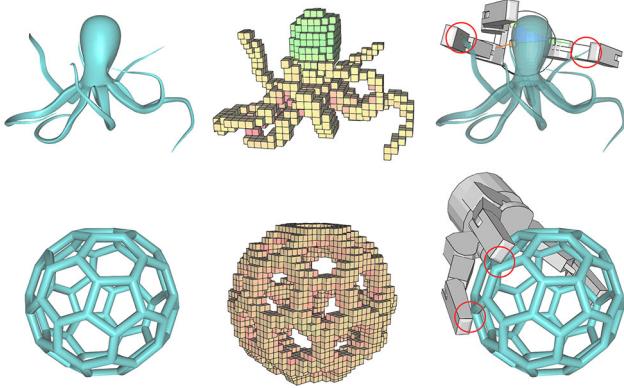


Fig. 21 Left branches (i.e., legs) of the OCTOPUS makes the target contact polyhedron has a very low chance to be reached by the hand due to unexpected hand-object collisions (see the red circles). Right contact score map of the CAGE provides little guidance for posing the hand for grasping, resulting in several unstable contacts (see the red circles)

8.1 Limitations and future work

First, there exist certain objects that cannot be handled by our grasp planning approach, e.g., objects with many branches [Fig. 21 (top)] or with frame structures [Fig. 21 (bottom)]. Second, object features such as holes in a bowling ball that are intentionally designed for grasping are not guaranteed to be captured since our approach considers low-level geometric features only when measuring grasp quality. We plan to address this issue by applying high-level semantic understanding on the objects to be grasped such as computing the tactile saliency [15]. Third, we plan to apply our grasp planning approach on real robotic systems to evaluate their performance in practice. Lastly, we plan to speed up our approach and apply it to real-time Augmented Reality applications such as understanding and grasping daily objects in augmented environments [10].

Acknowledgements This work is supported in part by Anhui Provincial Natural Science Foundation (1508085QF12), National Natural Science Foundation of China (61403357, 61672482, 11526212), and the One Hundred Talent Project of the Chinese Academy of Sciences.

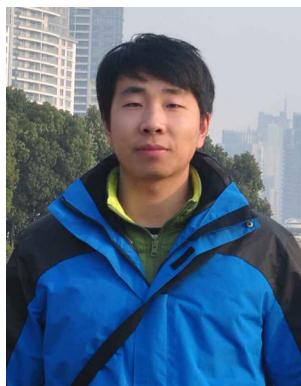
References

- Amor, H.B., Heumer, G., Jung, B., Vitzthum, A.: Grasp synthesis from low-dimensional probabilistic grasp models. *Comput. Anim. Virtual Worlds* **19**(3–4), 445–454 (2008)
- Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.* **22**(3), 181–193 (2006)
- Aydin, Y., Nakajima, M.: Database guided computer animation of human grasping using forward and inverse kinematics. *Comput. Graph.* **23**(1), 145–154 (1999)
- Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., Kuffner, J.: Grasp planning in complex scenes. In: IEEE-RAS International Conference on Humanoid Robots, pp. 42–48 (2007)
- Bohg, J., Morales, A., Asfour, T., Kragic, D.: Data-driven grasp synthesis—a survey. *IEEE Trans. Robot.* **30**(2), 289–309 (2014)
- Chalmeta, R., Hurtado, F., Sacristán, V., Saumell, M.: Measuring regularity of convex polygons. *Comput. Aided Des.* **45**(2), 93–104 (2013)
- Ciocarlie, M.T., Allen, P.K.: Hand posture subspaces for dexterous robotic grasping. *Int. J. Robot. Res.* **28**(7), 851–867 (2009)
- Ding, L., Ding, X., Fang, C.: 3D face sparse reconstruction based on local linear fitting. *Vis. Comput.* **30**(2), 189–200 (2014)
- Goldfeder, C., Allen, P.K., Lackner, C., Pelosof, R.: Grasp planning via decomposition trees. In: IEEE International Conference on Robotics and Automation, pp. 4679–4684 (2007)
- Güngör, C., Kurt, M.: Improving visual perception of augmented reality on mobile devices with 3D red–cyan glasses. In: the IEEE 22nd Signal Processing and Communications Applications Conference, pp. 1706–1709 (2014)
- Huebner, K., Ruthotto, S., Kragic, D.: Minimum volume bounding box decomposition for shape approximation in robot grasping. In: IEEE International Conference on Robotics and Automation, pp. 1628–1633 (2008)
- Kim, J., Iwamoto, K., Kuffner, J.J., Ota, Y., Pollard, N.S.: Physically based grasp quality evaluation under pose uncertainty. *IEEE Trans. Robot.* **29**(6), 1424–1439 (2013)
- Kry, P.G., Pai, D.K.: Interaction capture and synthesis. *ACM Trans. Graph. (SIGGRAPH)* **25**(3), 872–880 (2006)
- Kyota, F., Saito, S.: Fast grasp synthesis for various shaped objects. *Comput. Graph. Forum (Eurographics)* **31**(2), 765–774 (2012)
- Lau, M., Dev, K., Shi, W., Dorsey, J., Rushmeier, H.: Tactile mesh saliency. *ACM Trans. Graph. (SIGGRAPH)* **35**(4), 52:1–52:11 (2016)
- Li, Y., Fu, J.L., Pollard, N.S.: Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Trans. Vis. Comput. Graph.* **13**(4), 732–747 (2007)
- Li, Y., Saut, J.P., Pettré, J., Sahbani, A., Bidaud, P., Multon, F.: Fast grasp planning by using cord geometry to find grasping points. In: IEEE International Conference on Robotics and Automation, pp. 3265–3270 (2013)
- Miller, A.T., Allen, P.K.: GraspIt! A versatile simulator for robotic grasping. *IEEE Robot. Autom. Mag.* **11**(4), 110–122 (2004)
- Miller, A.T., Knoop, S., Christensen, H.I., Allen, P.K.: Automatic grasp planning using shape primitives. In: IEEE International Conference on Robotics and Automation, pp. 1824–1829 (2003)
- Nooruddin, F.S., Turk, G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.* **9**(2), 191–205 (2003)
- Park, Y.C., Starr, G.P.: Grasp synthesis of polygonal objects using a three-fingered robot hand. *Int. J. Robot. Res.* **11**(3), 163–184 (1992)
- Pollard, N.S., Zordan, V.B.: Physically based grasping control from example. In: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 311–318 (2005)
- Przybylski, M., Asfour, T., Dillmann, R.: Unions of balls for shape approximation in robot grasping. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1592–1599 (2010)
- Rimon, E., Burdick, J.: On force and form closure for multiple finger grasps. In: IEEE International Conference on Robotics and Automation, pp. 1795–1800 (1996)
- Roa, M.A., Suárez, R.: Grasp quality measures: review and performance. *Auton. Robots* **38**(1), 65–88 (2015)
- Rosales, C., Ros, L., Porta, J.M., Suárez, R.: Synthesizing grasp configurations with specified contact regions. *Int. J. Robot. Res.* **30**(4), 431–443 (2011)

27. Sahbani, A., El-Khoury, S., Bidaud, P.: An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.* **60**(3), 326–336 (2012)
28. Shen, C.H., Fu, H., Chen, K., Hu, S.M.: Structure recovery by part assembly. *ACM Trans. Graph. (SIGGRAPH Asia)* **31**(6), 180:1–180:12 (2012)
29. Ye, Y., Liu, C.K.: Synthesis of detailed hand manipulations using contact sampling. *ACM Trans. Graph. (SIGGRAPH)* **31**(4), 41:1–41:10 (2012)
30. Zhao, W., Zhang, J., Min, J., Chai, J.: Robust realtime physics-based motion control for human grasping. *ACM Trans. Graph. (SIGGRAPH Asia)* **32**(6), 207:1–207:12 (2013)
31. Zhou, Q., Panetta, J., Zorin, D.: Worst-case structural analysis. *ACM Trans. Graph. (SIGGRAPH)* **32**(4), 137:1–137:12 (2013)



Peng Song is an Associate Professor in the School of Computer Science and Technology, University of Science and Technology of China. He received his B.S. degree from the Harbin Institute of Technology in 2007, the M.S. degree from the Harbin Institute of Technology (Shenzhen) in 2009, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2013. His research interests lie in computer graphics, computer vision, and human-computer interaction. His research works could be found at his research Website: <http://staff.ustc.edu.cn/~songpeng>



Zhongqi Fu is currently pursuing the Master's degree in the School of Mathematical Sciences, University of Science and Technology of China. His research interests lie in geometric modeling and computer graphics.



Ligang Liu is currently a Professor at the School of Mathematical Sciences, University of Science and Technology of China. He received his B.Sc. (1996) and his Ph.D. (2001) degrees from Zhejiang University, China. Between 2001 and 2004, he worked at Microsoft Research Asia. Then, he worked at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include digital geometric processing, computer graphics, and image processing. He serves as the associated editors for journals of the IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Graphics and Applications, Computer Graphics Forum, Computer Aided Geometric Design, and The Visual Computer. He served as the conference co-chair of GMP 2017 and the program co-chairs of CVM 2016, SGP 2015, and SPM 2014. His research works could be found at his research Website: <http://staff.ustc.edu.cn/~lgliu>