Grasping Objects with Holes: A Topological Approach

Florian T. Pokorny, Johannes A. Stork and Danica Kragic

***Abstract*— This work proposes a topologically inspired approach for generating robot grasps on objects with 'holes'. Starting from a noisy point-cloud, we generate a simplicial representation of an object of interest and use a recently developed method for approximating shortest homology generators to identify graspable loops. To control the movement of the robot hand, a topologically motivated coordinate system is used in order to wrap the hand around such loops. Finally, another concept from topology – namely the Gauss linking integral – is adapted to serve as evidence for secure caging grasps after a grasp has been executed. We evaluate our approach in simulation on a Barrett hand using several target objects of different sizes and shapes and present an initial experiment with real sensor data.**

I. Introduction

A robot operating in a domestic environment often needs to identify suitable grasping configurations for objects with 'holes', such as cupboard door-handles, rubbish and grocery bags, cups etc. In this paper, we focus on the generation of grasp hypotheses for such objects – that is, in the language of algebraic topology, any object in $\mathbb{R}^3$ whose first homology group is non-trivial.

A popular approach [3] towards robotic grasping of a rigid object is based on the concept of *force-closure*. There, one needs to identify grasp configurations by considering both the forces exerted by the robot hand and the friction of the surfaces in contact. In this context, friction is usually approximated using the Coulomb friction model. The force analysis is then based on the exact determination of contact points and normals and relies on a reliable estimate of friction coefficients. A drawback of this approach to grasp selection is its imperative requirement of detailed knowledge of the grasped object's local geometry.

This paper explores an approach that is applicable to objects with holes, described in terms of algebraic topology by a non-trivial first homology group [7], [8]. Our approach can be seen as a method for synthesizing caging grasps on such objects – that is, grasps which restrict the object's movement to a small bounded set rather than fixating it completely. Such grasps are particularly appealing in many situations [9] since they allow to manipulate and move an object even without a *force-closed* configuration and without detailed estimates of friction coefficients and normal directions. We shall in fact only require a potentially noisy point-cloud representation of the object. Our methodology is in some sense dual to the classical *local* point-contact analysis because it uses *global* properties arising from topology in order to both identify suitable graspable regions and to guide the hand towards the final grasp configuration. In summary, the contributions of our work are:

a) An application – to our knowledge for the first time in a robotics context – of topological techniques developed in [8] in order to identify graspable regions of an object with 'holes'
b) The development of a new set of 'topology coordinates' (an approach similar to [34]) which is inspired by the concept of winding numbers and which enables motion planning and control in coordinates that describe the interaction between the object and the robot hand
c) An application of Gauss linking integrals to measure the linking between the robot fingers and an object
d) An evaluation of our approach on various objects and using a simulated Barrett hand.

The paper is organized as follows: in Sec. II, we briefly review related work and describe how a basis of the first homology group relates to loops on an object with holes. Furthermore, we discuss winding numbers and Gauss Linking integrals. Sec. III is dedicated to the topological representations of the grasped object and the robot hand and covers the theory of the motion generation procedure. We apply our method in Sec. IV and report on results.

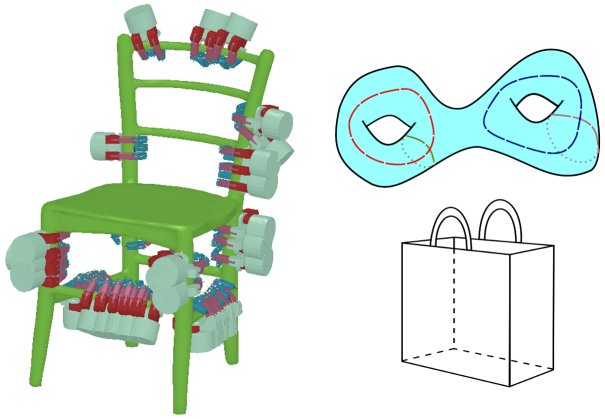


Fig. 1. Many every-day objects have holes. Grasping them with high Gauss linking between the robot hand and an interior loop cages the object and secures the grip. A surface $X \subset \mathbb{R}^3$ of genus two with four closed curves (dashed) generating $H_1(X)$ is displayed in top right. The figure on the bottom right shows a shopping bag with a two dimensional first homology group.

II. Background and Related Work

The problem of object grasping has been addressed frequently in the robotics community, resulting in a multitude

The authors are with the Computer Vision and Active Perception Lab, Centre for Autonomous Systems, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden, {fpokorny,jastork,dani}@kth.se

of approaches. The survey [3] reviews the theoretical foundations of analytic approaches to grasping, while a recent survey [29] provides also an analysis of empirical and data-driven methodologies. The grasping process depends on many factors, e.g., the physical properties of the object, its functionality, the scene context, the embodiment of the robot and the available sensory information. To cope with the underlying complexity of the problem, many approaches thus apply some form heuristic directly in the state space [29], [32], [26], [18], [6], [11], [20], [13], [14], [23].

By analysing a large set of human grasps, the authors of [30] determined that the first two principal components of the joint-space data were able to account for more than 80% of the variance of the grasps in their data-set. This notion of *postural synergies* has motivated many novel approaches [2], [12], [28]. While the use of synergies reduces the dimensions needed to represent the state of the robot hand, other authors have attempted to reduce the number of features needed for grasp analysis of the grasped object. Here, recent approaches have been based on approximating the object by matching shapes [19], [24] and finding a topological skeleton [27]. None of these methods however explicitly considers objects with holes as studied in our approach. In addition, most of the above work concentrates either on representing the hand's pose or the objects' features. In contrast to that, our approach attempts to build a non-linear low dimensional representation which is *based on concepts from topology* and which includes information about the object being grasped, the grasp position and the state of the robot hand. Instead of working with the force-closure condition which depends on the local geometry of an object and which is commonly used in grasp generation [3], [31], we shall consider a type of caging grasp motivated by the global topology of an object. Caging grasps have recently received renewed attention [9] for the execution of manipulation tasks and can sometimes allow for a less rigid approach to manipulation.

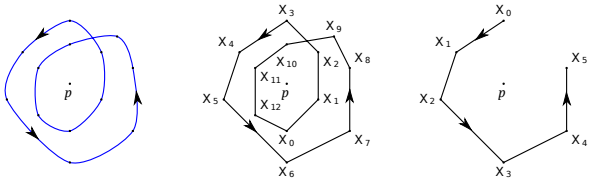
The philosophy of our approach is close to the work of [16] which investigated a state space representation for animated human characters based on the amount of linking between the character's limbs. There, a topological concept which applies in a strict sense only to closed curves was adapted to define a derived quantity called the *writhe matrix* to describe the characters' interaction. Unlike in our paper, the characters' motions are then prescribed by a series of key-frames given in the characters' original state space. We shall describe the target configuration by soft constraints and apply inverse kinematics to generate the hand's motion. In another related work [34], writhe matrices have been used to carry out motion planning and, in particular, it was shown that such matrices can be used to control the amount of wrapping of a multi-joint robot arm around a piecewise linear curve in the environment. While our approach is different from the above, we will also draw inspiration from algebraic topology.

A. Homology and loops

Algebraic topology [15] aims to classify topological spaces such as smooth manifolds up to certain continuous deformations such as homotopies. One of its main tools is the computation of homology which associates to every closed manifold a sequence of homology groups. In its simplest form, these homology groups $H_i(X)$ of a manifold X, with $i \in \{0, 1, \ldots, \dim(X)\}$, are vector spaces. The dimension of a homology group $H_i(X)$ can be considered as roughly representing the number of i-dimensional 'holes' in the manifold X. The case where X is a subset of $\mathbb{R}^3$ is of particular interest to us. In the special case where X is a smooth closed and connected surface, we have $\dim(H_0(X)) = \dim(H_2(X)) = 1$ and $\dim(H_1(X)) = 2g$, where $g \in \mathbb{N}_0$ denotes the genus of the surface.

A basis for the homology group $H_1(X)$ can be given by a set of equivalence classes of $\dim(H_1(X))$ closed curves lying on X. Two curves γ_1, γ_2 represent the same equivalence class in $H_1(X)$ if γ_1 can be continuously deformed into γ_2 by a one-parameter family of closed curves lying entirely in X [15]. The top right part of Fig.1 gives an example of a surface of genus 2, so that $\dim(H_1(X)) = 4$. Four closed curves representing basis elements for the first homology group are displayed as dashed lines. If we imagine this surface to be a filled solid double-torus rather than a hollow 2D surface, the corresponding topological space is in fact homotopy equivalent to the shopping bag displayed in Fig.1 and has a 2-dimensional first homology group since the two shortest curves can be contracted inside X to a point and do hence not represent non-zero elements in $H_1(X)$ anymore. Two generators for $H_1(X)$ can then be given by closed curves traversing the handles of the shopping bag in Fig.1. This discussion leads us to the notion of *handle* and *tunnel loops* as defined in [7]. Following their notation for a closed surface X, we denote by the *inside*, I, the union of X with the part of $\mathbb{R}^3$ which is contained inside X. Conversely, we denote the *outside* as $O = X \cup (\mathbb{R}^3 - I)$. A closed loop representing a non-trivial homology class in $H_1(X)$ is called a handle-loop if it is trivial in $H_1(I)$ — that is we can contract the loop to a point inside I — and non-trivial in $H_1(O)$, while a closed loop is called a tunnel loop if it is trivial in $H_1(O)$ but non-trivial in $H_1(I)$. In our surface in the top right of Fig.1, the two longer loops are hence tunnel-loops, while the two short ones which become trivial if we think of the shape as being filled out are handle loops. Tunnel loops hence 'tunnel' along a 'handle' of the object. The basic motivation presented in this paper is to apply grasps wrapping around such loops. Since we will however be working with noisy point-cloud data in this paper, it will not always be possible to reconstruct a clean water-tight mesh of the object of interest, making the determination of the 'inside' and 'outside' of an object difficult. Under these assumptions, we will hence treat all first homology generators as potentially graspable loops for the generation of grasp hypotheses in this work.

While methods for computing homology groups for a topological space X which is represented as a simplicial complex (such as a mesh) have existed since the invention of simplicial homology [15], this does not suffice in the context of grasping since any loop $\gamma : [0, 1] \to X$ representing a



(a) $w(\gamma) = 2$ (b) $w(\gamma) = 2$ (c) $w(\gamma) \approxeq 0.79$

Fig. 2. The winding number counts the number of full windings around a reference point in the smooth (a) and piecewise linear (b) case. In the piecewise linear non-closed case (c), the function is real-valued.

basis element of $H_1(X)$ is equivalent to any other loop $\hat{\gamma}$ that can be deformed into γ. We hence propose to use the recently developed method of [8] to instead identify a basis of $H_1(X)$ consisting of curves with approximately shortest length in their equivalence class. We shall in particular build simplicial complex representations from noisy point-clouds of objects and will then use the software ShortLoop [4] to identify shortest loops therein.

B. Winding numbers and Gauss linking integrals

Another topological concept that we incorporate in this work is the *winding number*. For a smooth closed curve $\gamma\colon [0,1] \to \mathbb{R}^2$ not containing the origin, the winding number $w(\gamma) \in \mathbb{Z}$ counts the number of times that γ wraps around the origin. The function $w(\gamma)$ is a topological invariant and does not change if we deform γ continuously without traversing the origin. If $\gamma(t) = (x(t), y(t))$ for $t \in [0,1]$, an explicit formula [17] for $w(\gamma)$ is given by

$$w(\gamma) = \frac{1}{2\pi} \int_0^1 \frac{\dot{y}(t)x(t) - \dot{x}(t)y(t)}{x(t)^2 + y(t)^2} \mathrm{d}t. \tag{1}$$

If γ is a closed piecewise-linear curve which connects the points $X_0, X_1, \ldots, X_{n-1}, X_n = X_0 \in \mathbb{R}^2$ by linear line segments, the winding number can still be computed by applying the above formula on each linear segment and summing the result [17]. The integral over each linear segment can easily be computed, leading to the formula

$$\begin{aligned} w(\gamma) = & \frac{1}{2\pi} \sum_{i=0}^{n-1} \left\{ \tan^{-1} \left(\frac{\langle X_{i+1}, X_{i+1} - X_i \rangle}{D_i} \right) \right. \\ & \left. + \tan^{-1} \left(\frac{\langle X_i, X_i - X_{i+1} \rangle}{D_i} \right) \right\}, \end{aligned} \tag{2}$$

where $D_i = (X_i)_1 (X_{i+1})_2 - (X_i)_2 (X_{i+1})_1$ and $\langle , \rangle$ denotes the standard inner product. Note that the winding number changes sign if the orientation of γ is reversed, i.e. for $\hat{\gamma}(t) \stackrel{\text{def}}{=} \gamma(1-t)$, we have $w(\hat{\gamma}) = -w(\gamma)$. Furthermore, we can define winding around any fixed point $p \in \mathbb{R}^2$ by choosing coordinates such that p corresponds to the origin. An example of such a curve is depicted in Fig.2(b).

If γ is not closed (so that $X_n \neq X_0$), $w(\gamma)$ defined by the above quantity takes values in $\mathbb{R}$ and is not a topological invariant in the strict sense anymore, but still provides a measure of the winding of γ around the origin. Fig.2(c) illustrates this concept for a non-closed curve.

For our work, it will be useful to consider $w(\gamma)$ as a function of the control points $X_0 = (x_0, y_0), \ldots X_n = (x_n, y_n)$. Observe that, as can be seen from Equation 1 and 2, $w(\gamma) : \mathbb{R}^{2(n+1)} \to \mathbb{R}$ is differentiable with respect to the coordinates of these control points as long as γ does not contain the origin and the computation of the Jacobian of this function is elementary.

Given two closed non-intersecting curves γ_1, γ_2 in $\mathbb{R}^3$, another useful invariant is the linking number $Lk(\gamma_1, \gamma_2) \in \mathbb{Z}$ which describes how the two curves are linked. If $Lk(\gamma_1, \gamma_2)$ is non-zero, the two curves cannot be 'pulled apart' without breaking the loops (see [15]). If $\gamma_1, \gamma_2 : [0,1] \to \mathbb{R}^3$ are smooth closed curves, the linking number can be computed using the Gauss linking integral:

$$\begin{aligned} & Lk(\gamma_1, \gamma_2) = \\ & \quad \frac{1}{4\pi} \int_0^1 \int_0^1 \langle \frac{\gamma_1(s) - \gamma_2(t)}{\|\gamma_1(s) - \gamma_2(t)\|^3}, \gamma_1'(s) \times \gamma_2'(t) \rangle \mathrm{d}s\mathrm{d}t. \end{aligned}$$

Just as in the case of the winding number, the above integral is well-defined and can be evaluated explicitly also in the case of two piecewise linear curves [21]. Similarly, we can define the above integral when γ_1, γ_2 are not necessarily closed and use the real-valued result as a measure for the amount of linking between two curves. This intuition is also used in [16], [34].

III. System Description and Implementation

We approach the problem of grasping objects with holes by exploring a departure from common local geometric representations of the robot's hand and the grasped object. In the following section we describe our topological object representation consisting of closed loops and our hand representation.

A. Topological object representation

We assume that the object being grasped is either already presented as a mesh or observed as a noisy point-cloud arising *e.g.* from RGBD-sensors or stereo vision. If point-cloud data is provided, we generate a Delaunay triangulation resulting in a tetrahedral mesh of the object's convex hull. Our heuristic method then removes all faces that contain an edge longer than $4cm$ since such edges connect rather distant points without additional evidence for data in between the vertices. The resulting simplicial complex X serves as input to the ShortLoop software [4], [8] to extract $k = \dim(H_1(X))$ piecewise-linear loops $\hat{\gamma}_1, \hat{\gamma}_2, \ldots, \hat{\gamma}_k$. The found loops represent generators of the first homology group and are approximately of shortest length within their homology class. Since we also know the physical dimensions of the robot hand, we can further reduce the number of candidate loops by rejecting loops $\hat{\gamma}_i$ of small length which are too small for the robot hand to grasp around. This results in a set $S = \{\gamma_1, \ldots, \gamma_m\}$ of piecewise linear curves representing potentially graspable loops. S is then used as a reduced topological representation of X. If S is empty, the data provides no detectable graspable loops and our algorithm does not apply.

B. Robot hand representation

For a given robot hand with l fingers, we select piecewise linear control curves, $C = \{\alpha_1, \alpha_2, \dots \alpha_l\}$, running from the fingertips through the joints to the tip of a respective opposable finger (see Fig. 3). For our algorithm to apply, it is necessary to be able to to identify at least one pair of opposable fingers. This is easily done for most current robot hands such as the case of a parallel gripper or for the Barrett, Schunk or DLR hand. Fig. 3 displays a possible choice for two control curves in case of a Barrett hand.

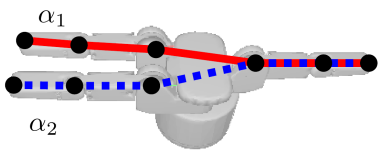


Fig. 3. Barrett hand model used in our experiments with control curves α_1 and α_2 indicated by lines. Both α_i contain the opposable finger to the right and span from finger tip to finger tip. The control points of these curves are given by the traversed joint-frame origins.

C. Topological control and motion planning

Let S and C denote the object and robot hand representation as defined above. S is hence a non-empty set of closed loops and C is a set of piecewise linear finger curves. We now consider how to generate a transition of the robot hand starting from an initial pose p_0 and yielding a grasp hypothesis around a loop $\gamma \in S$.

Any $\gamma \in S$ traverses an ordered set of points $X_0, X_1, \dots, X_s = X_0 \in \mathbb{R}^3$ by linear line segments. While we could choose to wrap the robot hand around any point on these segments, we decide to concentrate on the mid points $M_i = \frac{1}{2}(X_{i+1} - X_i)$ in this work. We form an orthonormal basis (T_i, V_{i1}, V_{i2}) to provide a target-centred coordinate system. The unit tangent vector of the segment, T_i, is given by $T_i = \frac{X_{i+1} - X_i}{\|X_{i+1} - X_i\|}$. We denote by V_{i1} the unit vector orthogonal to T_i and pointing towards the loop's centre $L = \frac{1}{s}\sum_{i=1}^{s} X_i$. The vector V_{i2} is then defined by a cross product: $V_{i2} = T_i \times V_{i1}$. Finally, we let Π_i denote the plane through M_i with normal direction T_i. We denote the orthogonal projection that maps $\mathbb{R}^3$ onto the plane Π_i by π_i. Fig. 4 elucidates the definitions.

For a fixed loop $\gamma \in S$ and segment index $i \in \{0, 1, \dots, s-1\}$ we can now define a constraint optimization problem. For all control curves $\alpha \in C$:

a) Maximize the winding of the projection $\pi_i(\alpha)$ around the point M_i and approach the object along the direction V_{i1}
b) Minimize the distance between the robot hand and the plane Π_i
c) Avoid collisions with the object being grasped.

In combination with the starting pose heuristic, described in section IV, these constraints define a transition that is initially similar to the notion of approach vectors [10]. The robot hand will have to approach the point M_i to increase winding numbers. However, the constraints enforce the adaptation of the finger configuration and the hand's pose to the target object's local shape around the loop.

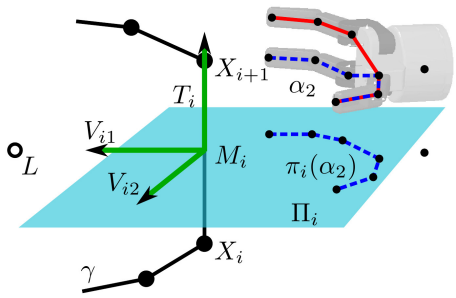


Fig. 4. The loop γ is grasped around its i-th segment's midpoint M_i. The control curve α_2 is projected orthogonally onto the plane Π_i which is spanned by the vectors V_{i1} and V_{i2}. For the construction of the orthonormal basis $(T_i, V_{i,1}, V_{i,2})$ from the normal T_i and the loop's centre L, see section III-C.

Since this work focuses on introducing new topological representations for grasping, we will describe here only a rather simple approach towards attaining the above goals using inverse kinematics and a task space mapping. We consider a given robot hand with $u \in \mathbb{N}$ degrees of freedom and define the state space $P = \mathbb{R}^3 \times SO(3) \times \mathbb{R}^u$, where the space $\mathbb{R}^3$ represents the position of the robot hand's base, $SO(3)$ its orientation and $\mathbb{R}^u$ the hand's degrees of freedom (such as joint angles). Let us fix a $\gamma \in S$ and a particular segment index i and let $C = \{\alpha_1, \dots, \alpha_l\}$. We consider a task space mapping, $\Gamma\colon P \to \mathbb{R}^{l+3}$ that maps poses $p \in P$ from the robot's state space into a topologically inspired coordinate system and where

$$\Gamma(p) = (w_1(p), w_2(p), \dots, w_l(p), d(p), c(p), r(p))^T. \quad (3)$$

Here, $w_j(p)$ denotes the winding number of the projected curve $\pi_i(\alpha_j) \subset \Pi_i$ around the point $M_i \in \Pi_i$. Furthermore, we augment the winding information with the function $d\colon P \to \mathbb{R}$, denoting the mean distance between the vertices traversed by any of the control curves $\alpha \in C$ and their own image under π_i. We model the collision constraint by applying a collision potential $c\colon P \to \mathbb{R}$ that is defined to be zero if the robot is at a distance of more than $1cm$ from the object and which, starting at 0, then smoothly increases as the distance to an obstacle approaches zero. Finally, we add a function, $r\colon P \to \mathbb{R}$, which measures the distance between the projection of the palm's centre onto Π_i and the line through M_i and V_{i1}. Minimizing d now yields collision-free configurations while minimizing c lets the hand approach the plane Π_i. Finally, a minimization of r yields an approach such that the hand is oriented towards the focus point M_i.

The Jacobian $J\Gamma$ for Γ can now be computed, where the derivatives of the functions w_i can be computed directly from Equation 2 or by numerical approximation. We shall then apply a simple Jacobian transpose based controller to approach a task-space goal state $g \in \mathbb{R}^{l+3}$. I.e., given a pose $p_t \in P$, we iterate by updating the state by

$$p_{t+1} = p_t + \alpha(J\Gamma(p_t))^T(g - \Gamma(p_t)) \quad (4)$$

for some small fixed $\alpha > 0$. In our experiments, we shall define the goal state by setting the last three coordinates to zero and by setting the goal winding numbers of the first l coordinates as the current winding plus a fixed positive offset $\delta > 0$.

Procedure 1 The generation of grasp hypotheses is based on a mesh generated from point-cloud data. All object loops are extracted and a priori incompatible loops are rejected. Segments of remaining loops are subject to a grasp approach and a heuristic is used to identify an initial pose p_0. The Jacobian transpose controller terminates upon unsubstantial improvement. The symbols below are defined in section III.

Input: Object point data D, robot hand representation C.
Output: Set of grasp hypotheses G.

```
G ← ∅
S ← ShortLoop(D)                                   ▷ See section III-A
S ← Remove-Incompatible-Loops(S)
for all γ ∈ S do
    for all segments (X_i, X_{i+1}) ⊂ γ do
        p_0 ← Starting-Pose-Heuristic(X_i, X_{i+1})
                                                   ▷ See section IV
        if p_0 ≠ ∅ then
            Γ ← Task-Space-Mapping(C, Π_i, π_i)
                                                   ▷ See section III-C
            G = G ∪ Jacobian-Transpose-Control(Γ, p_0)
                                                   ▷ See sections III-C, IV
        end if
    end for
end for
```

In summary, the complete grasp generation procedure hence consists of the following steps: first, all object loops are extracted. Loops whose length is larger than a threshold are then considered segment-wise, and for each segment, a heuristic initial pose of the robot hand is generated, as explained in section IV. Finally, the task-space mapping function Γ, together with the regarded segment and the initial pose, instantiate a constraint optimization problem that is solved via the Jacobian transpose controller described in Eq. (4). See Procedure 1 for a pseudo-code description.

IV. Experimental Evaluation

We now describe our experimental setup and results. First, we provide details on how a set of piecewise-linear loops, $S = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ is generated for our test objects, then we comment on the robot representation. Next, we explain our heuristic selection of initial poses and give details on which loops and initial poses are rejected a priori. Finally, we present results on the generated grasp hypotheses on objects of different sizes and shapes, as well as on real sensor data.

We will consider the objects displayed in Fig. 5 for our experiments. We generated these point-clouds from an underlying mesh which we obtained from [1]. Next, we sampled a maximum of 2100 uniform points from each mesh and added zero mean Gaussian noise with standard deviation $0.3cm$. While the ShortLoop software [4] can construct simplical complexes from point-cloud data, we decided to provide it directly with a simplical complex to speed up the computation. To create such a complex from the point-cloud data, we generate a filtered Delaunay triangulation by rejecting faces that contain edges of length larger than $4cm$ as previously described. As can be seen in the bottom row of Fig. 5 this results in a realistic reconstruction of our objects. Finally, we determine an approximately shortest homology basis of the first homology group using the ShortLoop software. For the given objects, the mesh creation and loop detection took no more than $10s$ per object on a current Intel i7 CPU laptop. We use the libORS [33] robot simulator to simulate a four degree of freedom Barrett hand as depicted in Fig. 3. The two control curves α_1 and α_2 connect the finger tips of the two rotatable fingers (on the left in Fig. 3) to the finger tip of the non-rotatable 'thumb' of this robot hand. The curves run from tip to tip traversing joint positions. To define the collision potential, we use the distance to the original object mesh.

Since we do not address planning in this work, we employ a simple heuristic for setting the initial pose $p_0 \in P$ of the robot hand: for each loop γ and each linear segment (X_i, X_{i+1}) on γ, we position the robot hand $50cm$ away from the point M_i along the half-line through M_i defined by the vector $-V_{i1}$. The hand is then oriented towards M_i and the fingers are positioned along the plane Π_i such that the initial pose yields a positive winding number for the two control curves.

If the line between the hand position and M_i intersects the filtered Delaunay mesh at a position further than $30cm$ away from M_i, we assume that the approach direction is rendered unusable by an obstacle and dismiss the particular initial configuration. This occurs for example when the robot attempts to grasp the blue loop of the chair in Fig. 5 from above. Fig. 8 depicts accepted starting positions.

Let us now come to the definition of the task goal state $g \in \mathbb{R}^{l+3}$. The last three coordinates of g are set to zero, representing the distance to Π_i, collision, and approach constraints. For the remaining l winding number constraints, we wish to maximize each winding number. Since the maximal obtainable winding numbers between the control curves and the target loop are however unknown, we follow the method in [5] and select as the target for the state p_t at iteration t the current task winding numbers plus a fixed increase. Eq. (4) now becomes $p_{t+1} = p_t + \alpha(J\Gamma(p_t))^T(g_t - \Gamma(p_t))$ with

$$g_t = (w_1(p_t), w_2(p_t), \ldots, w_l(p_t), d(p_t), c(p_t), r(p_t))^T + (\delta, \delta, \ldots, \delta, 0, 0, 0)^T \quad (5)$$

for $\delta = 0.8$. The magnitude of change is then clamped in each update step. We stop the Jacobian transpose control procedure if the increase in the winding numbers is insignificant for a period of 200 control iterations. Finally, we tighten the grasp by an auto-grasp procedure [25]. The robot hand is moved forward along its principal direction until contact occurs. The fingers are then closed one-by-one until they have contact with the object or the hand itself.

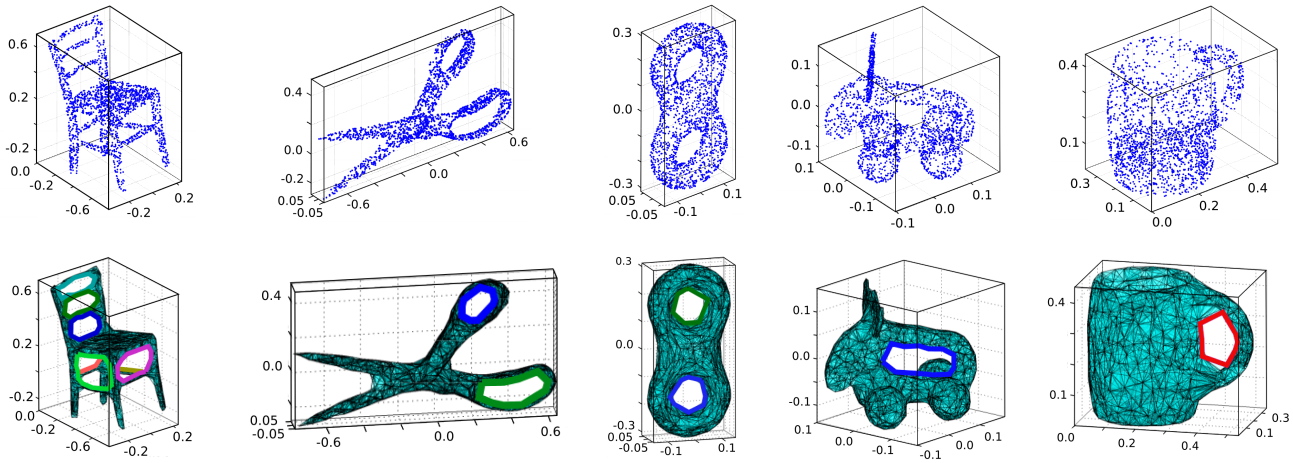


Fig. 5. Point-cloud data for our test objects is displayed in the top row. A resulting filtered Delaunay triangulation and detected loops are displayed in the bottom row. The point clouds were sampled from meshes and zero-mean Gaussian noise with standard deviation $0.3cm$ was added to simulate real point-cloud sensor data. The units in all plots are metres.

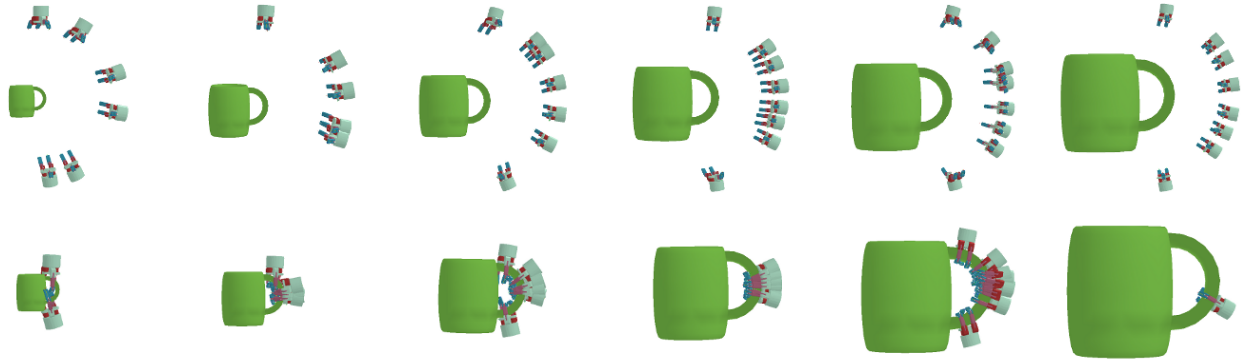


Fig. 6. Grasp hypothesis generation on an object in different sizes shows that our approach reliably produces similar hypotheses. The object scale varies in steps of 20% from 40% to 140%. All accepted initial poses are positioned similarly across sizes (top). Considering the resulting grasp hypotheses (bottom) such that at least one control curve has a Gauss linking higher than 0.75 with the loop, we observe that some grasp regions are shared amongst the different object sizes (bottom). For the smallest size, near-horizontal grasps are missing since our final auto-close procedure results in a collision between the fingers and the cup's body before being able to fully close around the cup's handle. For the largest size, the handle is so big and the hand cannot wrap around it completely.

To test the variance in placement and quality of the generated grasp hypotheses with the scale of an object, we first considered the cup displayed in Fig.6. The cup is scaled from 40% to 140% in steps of 20% each and approximately 1000 points are sampled on the cup's surface without additional noise. Only a single loop describing the cup's handle is detected in the refined Delaunay triangulation for all sizes. The number of generated initial poses varies from 11 in the 60%-case to 15 in the 100% and 140%-cases. Between 6 and 8 initial poses are rejected because of obstruction by the cups body, see Fig. 7 (top). For all sizes, the initial poses are positioned on the handle side of the cup in a similar manner and allow for a direct approach as depicted in Fig. 6 (top). The most common place of the grasp hypotheses with high Gauss linking (0.75 or higher) is the middle of the handle-part and is shared amongst the different sizes. The relation between the fixed size of the robot hand and the varying size of the object is reflected as a trend in the distribution of the resulting Gauss linking. Fig. 7 (bottom) shows the maximal Gauss linking of the two control curves in the final pose. In case of the largest cup, the handle-part is enlarged to an extend that the robot hand can barely encompass it completely. While only one grasp hypothesis with Gauss linking larger than 1.0 is produced, this does however not necessarily imply that most of the other grasp hypotheses do not cage the object. In case of the smallest cup, the robot hand is not able to fit the two fingers through the loop completely and therefore is prevented from obtaining a high linking number with the loop. The generated grasp hypotheses with Gauss linking higher than 0.75 are shown in Fig. 6 (bottom).

Let us now consider the five objects from Fig. 5 for grasp hypothesis generation and let us investigate how the shape of these objects influences our approach. Note that we do only encounter tunnel loops in our computations, since - for our examples - our filtered Delaunay triangulation (e.g. for

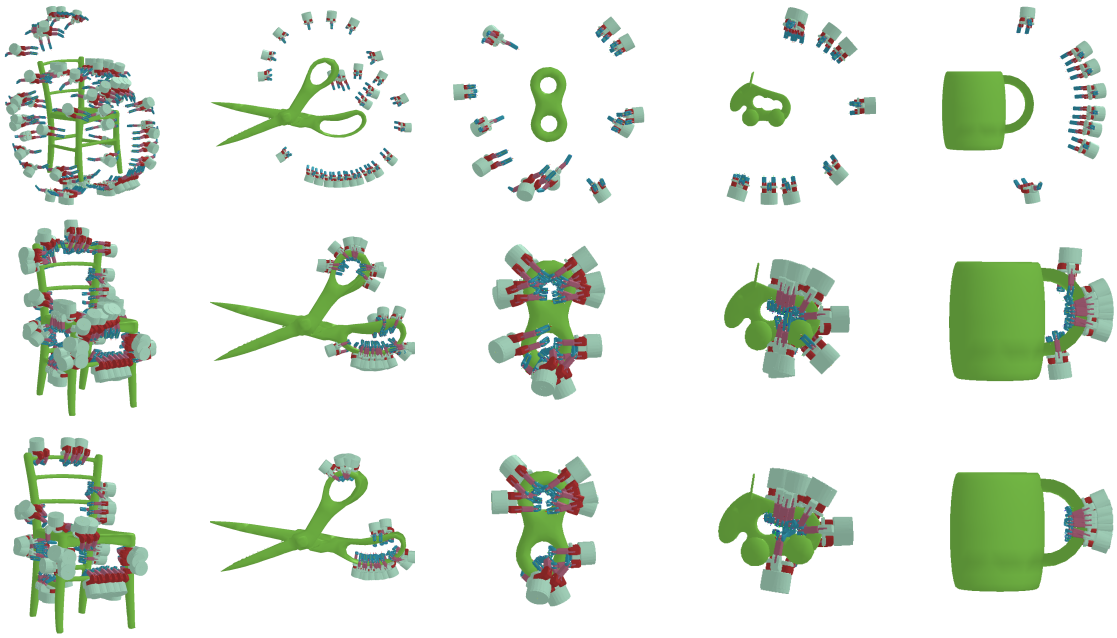


Fig. 8. Generated initial positions for different objects (top), all resulting final grasp configurations (middle) and final grasp configuration with Gauss linking 0.75 or larger (bottom).

the chair model) produces a 'filled' solid complex which contains solid tetrahedra modelling the interior of the object. The number of detected loops is 7, 2, 2, 1 and again 1 for the chair, scissors, double torus (d.torus), toy and cup, respectively. The initial poses are displayed in Fig. 8 (top). The number of generated initial poses varies with the number of loops and is 165, 15, 15, 38 and 16 for the objects in the above order. Many initial poses are rejected for the chair since the hand's path is obstructed by the chair's geometry. Considering the distribution of the maximal Gauss linking, we can see that all objects but the toy allow for a large share of the grasp hypotheses to assume a Gauss linking higher then 1.0. The complicated local geometry of the toy is reflected in its Gauss linking distribution as can be seen in Fig. 9 (bottom). In the chair experiment, the robot hand cannot always reach the target loop segment and gets stuck at other parts of the object.

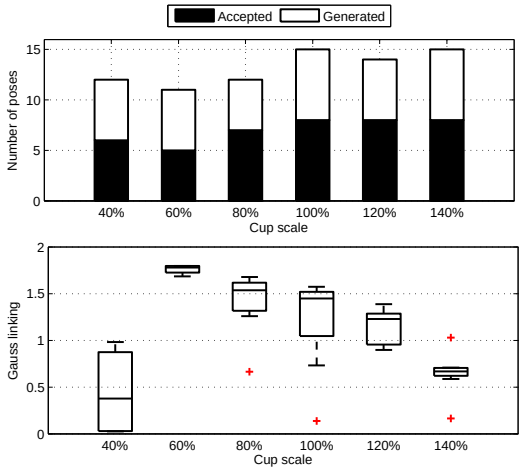


Fig. 7. While the ratio of generated initial poses to rejected initial poses is stable in for different sizes of the cup object (top), the distribution of the Gauss linking in the resulting grasp hypotheses varies (bottom). The trend shows that the 60%-scaled cup fits the robot hand best. The edges of the box-plot show the 25th and 75th percentiles.

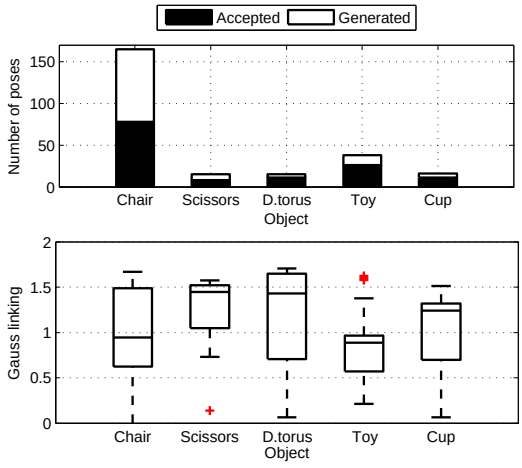


Fig. 9. Results for the grasping experiment with five different objects. The number of generated initial poses is the highest for the chair since it has 7 loops (top). All objects allow for grasps with high Gauss linking, but the complicated shape of the toy is clearly reflected in the distribution. The edges of the box-plot show the 25th and 75th percentiles (bottom).

Let us now consider an initial experiment with real data. We consider segmented partial view point-cloud data from [22]. An image of an object is shown in Fig. 10 together with an associated point-cloud comprising 3213 data points. We generated a filtered Delaunay mesh and computed the shown loop and the resulting grasp hypotheses (see Fig. 10).

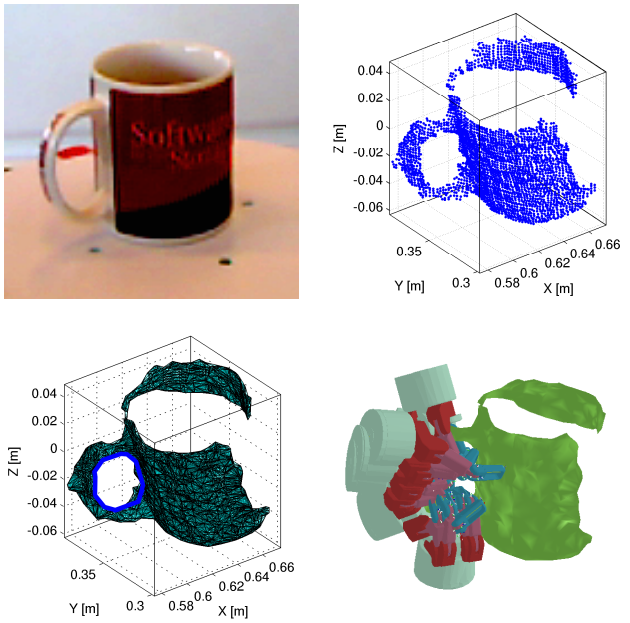


Fig. 10. Coffee mug (top left) and corresponding point-cloud (top right), extracted loops and filtered Delaunay mesh from real sensor (bottom left) as well as nine resulting grasp hypotheses with Gauss linking larger equal to 1.0 are shown (bottom right).

To execute our grasp hypothesis generation approach in simulation, we compute the collision potential from the filtered Delaunay mesh and scale the mesh by a factor of 2.5 beforehand. This results in 13 grasps 9 of which have a Gauss linking of at least 1.0 with the handle loop.

V. Conclusion

In this paper, we have developed a grasp state representation and control method for grasping objects with 'holes', drawing inspiration from several topological concepts such as homology groups, winding numbers and Gauss linking integrals. We have applied our representation to successfully generate grasps on objects of different sizes and shapes in simulation. Fig. 10 presents an initial experiment using data generated with a real sensor, but a more thorough analysis with more object categories, view angles and with different noise levels will need to be performed. In the future, we also intend to demonstrate our approach on a real robot system and would like to incorporate a planning framework such as rapidly exploring random trees.

VI. Acknowledgements

This work was supported by the EU project FLEXBOT (FP7-ERC-279933) and the Swedish Foundation for Strategic Research.

References

[1] AIM@SHAPE. The AIM@SHAPE Repository.
[2] A. Bicchi, M. Gabiccini, and M. Santello. Modelling natural and artificial hands with synergies. *Phil. Trans. R. Soc.*, 2011.
[3] A. Bicchi and V. Kumar. Robotic Grasping and Contact: A Review. In *IEEE ICRA*, 2000.
[4] O. Busaryev, T.K. Dey, J. Sun, and Y. Wang. ShortLoop Software for Computing Loops in a Shortest Homology Basis. Software, 2010.
[5] S. R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation, 2004.
[6] M.T. Ciocarlie. *Low-Dimensional Robotic Grasping: Eigengrasp Subspaces and Optimized Underactuation*. PhD thesis, Columbia University School of Arts and Sciences, 2010.
[7] T.K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3D models. In *ACM SIGGRAPH*, pages 1–9, 2008.
[8] T.K. Dey, J. Sun, and Y. Wang. Approximating loops in a shortest homology basis from point data. In *ACM SoCG*, pages 166–175, 2010.
[9] R. Diankov, S. S. Srinivasa, D. Ferguson, and J. Kuffner. Manipulation planning with caging grasps. In *Humanoids*, pages 285–292, 2008.
[10] S. Ekvall and D. Kragic. Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning. In *IEEE ICRA*, pages 4715–4720, 2007.
[11] S. El-Khoury and A. Sahbani. Handling Objects By Their Handles. In *Workshop at IEEE/RSJ IROS*, 2008.
[12] M. Gabiccini, A. Bicchi, D. Prattichizzo, and M. Malvezzi. On the Role of Hand Synergies in the Optimal Choice of Grasping Forces. *Auton. Robot*, 2011.
[13] J. Glover, D. Rus, and N. Roy. Probabilistic Models of Object Geometry for Grasp Planning. In *RSS*, 2008.
[14] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp Planning Via Decomposition Trees. In *IEEE ICRA*, pages 4679–4684, 2007.
[15] A. Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.
[16] E.S.L. Ho and T. Komura. Character Motion Synthesis by Topology Coordinates. *Comput. Graph. Forum*, 28(2):299–308, 2009.
[17] K. Hormann and A. Agathos. The Point in Polygon Problem for Arbitrary Polygons. *Computational Geometry. Theory and Applications*, 20(3):131–144, 2001.
[18] K. Hsiao, S. Chitta, M. Ciocarlie, and E.G. Jones. Contact-Reactive Grasping of Objects with Partial Shape Information. In *IEEE/RSJ IROS*, pages 1228–1235, 2010.
[19] K. Huebner, S. Ruthotto, and D. Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *IEEE ICRA*, pages 1628–1633, 2008.
[20] I. Kamon, T. Flash, and S. Edelman. Learning to Grasp Using Visual Information. In *IEEE ICRA*, pages 2470–2476, 1994.
[21] K. Klenin and J. Langowski. Computation of writhe in modeling of supercoiled DNA. *Biopolymers*, 54:307–317, 2000.
[22] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *IEEE ICRA*, pages 1817–1824, 2011.
[23] Y. Li and N.S. Pollard. A Shape Matching Algorithm for Synthesizing Humanlike Enveloping Grasps. In *IEEE/RAS Humanoids*, pages 442–449, Dec. 2005.
[24] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic Grasp Planning Using Shape Primitives. In *IEEE ICRA*, pages 1824–1829, 2003.
[25] A.T. Miller and P.K. Allen. Graspit! A Versatile Simulator for Robotic Grasping. *Robotics Automation Magazine, IEEE*, 11(4):110–122, 2004.
[26] M. Przybylski and T. Asfour. Unions of Balls for Shape Approximation in Robot Grasping. In *IEEE/RSJ IROS*, pages 1592–1599, 2010.
[27] M. Przybylski, T. Asfour, and R. Dillmann. Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In *IEEE/RSJ IROS*, pages 1781–1788, 2011.
[28] J. Romero, T. Feix, C. H. Ek, H. Kjellström, and D. Kragic. Extracting Postural Synergies for Grasping. *RSS*, 2012.
[29] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 2011.
[30] M. Santello, M. Fl, and J. F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 1998.
[31] J.-P. Saut and D. Sidobre. Efficient models for grasp planning with a multi-fingered hand. *Robotics and Autonomous Systems*, 2012.
[32] J. Seo, S. Kim, and V. Kumar. Planar, Bimanual, Whole-Arm Grasping. In *IEEE ICRA*, pages 3271–3277, 2012.
[33] M. Toussaint. libORS Open Robotics Simulator. Software, 2012.
[34] D. Zarubin, V. Ivan, M. Toussaint, T. Komura, and S. Vijayakumar. Hierarchical Motion Planning in Topological Representations. In *Proceedings of Robotics: Science and Systems*, 2012.