Interactive Grasping for High-genus Objects using Configuration Space Learning

Hao Tian[1], Changbo Wang[1], Dinesh Manocha[2] and Xinyu Zhang[*,1]

***Abstract*— We present an interactive grasping algorithm for manipulating high-genus complex rigid objects, without assuming the geometric shape of the object. We use machine learning and particle swarm optimization to automatically compute the grasp poses for a given object and a multi-fingered hand. Our algorithm can autonomously search for feasible grasp poses and determine their grasp stability. We express non-penetration and physically-based conditions as constraints in a grasp stability function. We test our algorithm in simulation using a few high-genus complex objects with different sizes and shapes. Our algorithm is applicable to a wide range of complex objects and multi-fingered robotic hands.**

I. Introduction

Robotic grasping has a wide range of applications in automation and robotics. Given an object and robotic hand, grasping refers to the problem of finding a set of stable hand poses, under which the object can be firmly grasped. Many techniques have been proposed to synthesize or plan grasps. Analytic approaches take into account the laws of physics and address the problems of contact computation, force-closure, grasp feasibility, grasp stability [3], [10], [12], [21], etc. Data-driven approaches gather grasping knowledge by sampling grasp configurations in virtual simulations or recording how a human performs grasping tasks [4], [8], [14], [26], [38].

Although these grasp methods are promising, they have not yet been demonstrated to be reliable enough to be used for interactively manipulating complex objects, especially for objects with high-genus. Consider the fact that semantically modeling the geometric shape of complex objects is non-trial and the degree of freedom of a robotic hand is very high, grasping complex objects is a very challenging problem and it still remains an active research area in robotics. Moreover, many factors have to be taken into account, including collision handling, contact friction and physical stability [20], [32], to ensure stable grasp poses. To achieve interactive performance, some approaches approximate the object shapes with simple primitives such as spheres, cones, cylinders, and boxes [16], [24], [30], or superquadrics (SQ) [13]. Some approaches pre-process the object shape using topological analysis [28] and gain grasping knowledge from human expertise [9]. A few approaches directly sample configuration space to reduce complexity [6], [33], [34].

Main Results: We present an interactive grasping algorithm for manipulating high-genus complex rigid objects, without assuming the geometric shape of the object. We use machine learning and particle swarm optimization to automatically compute the grasp poses for a given object and a multi-fingered hand. Our algorithm can autonomously search for feasible grasp poses and determine their grasp stability. Our algorithm is applicable to a wide range of complex objects and robotic hands, as shown in Figure 1.

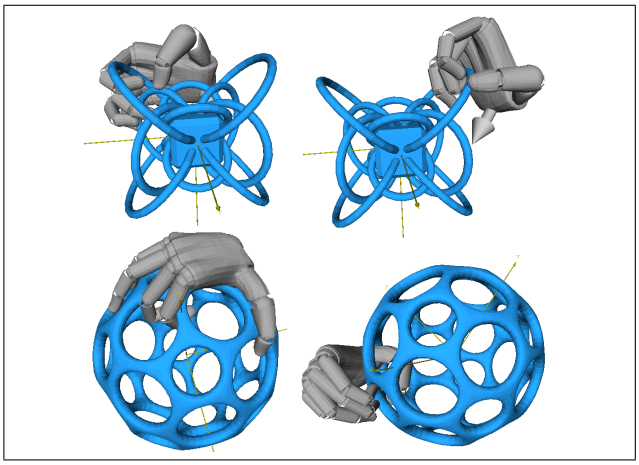


Fig. 1. Learning grasp poses for a five-fingered robotic hand and high-genus complex objects.

II. Related Work

In this section, we give a brief overview of prior work on robotic grasping.

Analytic Approaches: These techniques account for laws of physics to compute feasible stable grasps and have been well-studied in dynamics and robotics [3]. Markenscoff et al. [21] used grasp stability analysis to balance various forces acting on objects. Due to the complexity of stability analysis, most methods are limited to hands with low degrees-of-freedom (DOFs), and may not work well on high-DOF hands. Force-closure and form-closure [25] were designed to guarantee the maintenance of a set of contacts. Some grasp quality measures were suggested in the computation of optimal grasps [35]. One widely used measure was formulated as the minimum distance between the origin of the wrench space and the boundary of a grasp wrench set. A linear friction

[1]Hao Tian, Changbo Wang and Xinyu Zhang are with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China.

[2]Dinesh Manocha is with the Department of Computer Science and Electrical & Computer Engineering, University of Maryland at College Park, MD, USA.

[*]Xinyu Zhang is the corresponding author. xyzhang@sei.ecnu.edu.cn.

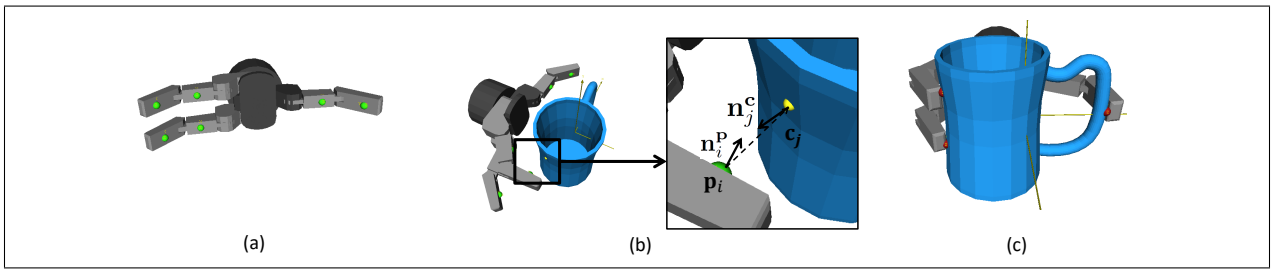


Fig. 2. Grasp pose and associating contacts. (a) A grasp pose and the pre-defined contact locations on the hand (green points). (b) The pre-defined point and its corresponding closet point on the object surface (yellow point). (c) Contacts (red points) on the object surface corresponding to a grasp pose.

model was proposed in [2]. The grasp wrench set was treated as a polytope and the minimum distance approximated using all facets of the polytope [22]. However, the grasp wrench set is computed using Minkowski sum and the performance is slow for interactive applications [40]. Some approaches ensured force-closure by finding contact points on object surfaces [10], [18]. Fischer et al. [12] generated a number of feasible grasp candidates heuristically. Most of these methods are not suitable for interactive applications.

Data-driven Approaches: These techniques gain grasping knowledge via recording how a human performs grasping and then synthesizing grasp motions [4]. Grasping data can be collected using digital gloves [36], optical devices [39] or computer vision algorithms [1]. However, large scale data collection can be time-consuming and requires lots of human volunteers. Some automatic algorithms were proposed to collect grasp data [8], [14], [26], [38]. Diankov [8] determined grasp candidates using the bounding box of an object in conjunction with surface normals. Zacharias et al. [38] computed grasp maps via contact points on object surfaces. Goldfeder et al. [14] constructed the grasp database using optimal grasps generated using simulated annealing. Pelossof et al. [26] applies machine learning to grasping arbitrary objects. The resulting algorithm pre-defined the sampling positions and orientations evenly around the simplified object. Given pre-recorded motion data, there is considerable work on grasping synthesis [39]. Grasping rules [23] were predefined for basic geometric primitives and could be used for other shapes. Li et al. [19] used shape matching techniques to find grasp candidates. However, for a new object, visual artifacts like inter-penetrations become severe due to shape differences. Posture refinement is required to decrease these visual artifacts. Grasping and manipulating motions can be synthesized by incorporating physical constraints [17], [29], [37], [39]. A learning approach [15] was presented for grasping objects. A Gaussian mixture model was trained to model the distribution of hand postures for a specific object. This model was then used to find finger joints for a given hand pose. In contrast to these methods, our approach involves no data gathering and can be directly applied to any complex object.

Grasping Complex Objects: Many techniques have been proposed to grasp complex objects by approximating the object shapes with a set of primitives. Miller et al. [24] simplified an object as a set of shape primitives, such as spheres, cylinders, cones, and boxes, they use a set of rules to generate a set of grasp starting positions and pregrasp shapes that can then be tested on the object model. Goldfeder et al. [13] decomposed a 3D model into a superquadric "decomposition tree" which can be used to prune the large configuration space into a subspace that is likely to contain many good grasps. Huebner et al. [16] decomposed a point cloud into a set of boxes. The simple geometry of a box reduces the number of potential grasps significantly. In [9], the graspable parts are learned based on the parameters of superquadrics fitted to segmented parts of the object and using human expertise. In [28], topological concepts such as homology groups, winding numbers and Gauss linking integrals homology groups were investigated to generate caging grasps on objects with holes.

III. Overview

Assuming that a multi-fingered hand consists of k joints and the joint variables are $\Theta = \{\theta_1, \theta_2, \cdots, \theta_k\}$. Due to the high number of finger joints, a grasp pose is a high-dimensional vector, which belongs to the grasp space constructed using a multi-fingered hand and an object to be grasped. A grasp space corresponds to a set of stable grasp configurations at which an object can be firmly grasped. To grasp an object, a multi-fingered robotic hand must have multiple contacts with the object. As shown in Figure 2, a grasp pose corresponds to a few contacts between the hand and the object. The predefined points on the hand is denoted by $\mathbf{p}$ and its normal is denoted by $\mathbf{n}_i^{\mathbf{P}}$. Their corresponding closet points on the object surface found during optimization process is denoted by $\mathbf{c}$ and its normal are denoted by $\mathbf{n}_j^{\mathbf{c}}$. The contacts on the object surface are shown as red points.

Given a multi-fingered hand, our goal is to autonomously learn stable grasp poses for high-genus complex objects. Our approach makes no assumptions about the shape of an object to be grasped, but we assume that the exact shape of the rigid object is known. Therefore, we propose an algorithm to approximate the grasp space by using a combination of machine learning and optimization techniques. We first compute an approximation of the grasp space using support vector machine. Then, we find potential grasp poses using

particle swarm optimization. These two steps are perform during pre-processing stage. Interactive grasping for complex objects can achieved at runtime stage.

IV. Configuration Space Learning

In this section, we present our pre-processing algorithms that learn configuration space.

A. Sampling Configuration Space

We first randomly sample the configuration space and compute the collision state for each sample configuration using a discrete collision detection algorithm. There are two possible collision states: in-collision or collision-free, which correspond to a scenario in which the robotic hand collides with the object and a scenario in which it does not, respectively. Given a set of samples in configuration space, we use an SVM technique to train a binary classifier to approximate the configuration space and the resulting decision boundary separates all in-collision configurations from the collision-free configurations. Grasp poses lie on the decision boundary that corresponds to a scenario in which the robotic hand touches the object.

To incorporate physically-based constraints, we define the following *energy function* to model the shape fitting and grasping stability of a grasp pose. Suppose we have a set of pre-specified points on the hand [5]. The energy function is formulated as

$$\sum_i \sum_j (\omega_1 \|\mathbf{p}_i - \mathbf{c}_j\| + \omega_2 (1 - \mathbf{n}_i^{\mathbf{p}} \cdot \frac{\mathbf{p}_i - \mathbf{c}_j}{\|\mathbf{p}_i - \mathbf{c}_j\|})) + \omega_3 \log \epsilon^{-1}, \quad (1)$$

where the first term $\|\mathbf{p}_i - \mathbf{c}_j\|$ is the distance between the ith pre-defined point $\mathbf{p}_i$ on the hand and its corresponding closest point $\mathbf{c}_j$ on the object surface. The second term is based on the angle between the surface normal $\mathbf{n}_i^{\mathbf{p}}$ at $\mathbf{p}_i$ and the vector from $\mathbf{p}_i$ to $\mathbf{c}_j$. The last term relates to grasp quality [11] and $\epsilon \in (0, 1]$ when the grasp is stable. A small ϵ indicates a relatively small external disturbance that can break a grasp's stability. ω_1, ω_2, and ω_3 are the weights for the three terms, respectively.

We minimize the energy function with respect to the hand pose and joint variables Θ using *particle swarm optimization* [7] and then determine the stable grasps on the object. This formulation is highly non-linear [27] and small changes in either hand position or finger postures can drastically alter the quality of the resulting grasp. The stochastic nature of particle swarm optimization makes it a particularly good choice for solving the problem. Since a new configuration used in particle swarm optimization is generated as a neighbor of the current configuration, we can avoid collisions during the sampling process. In addition, the computation of an energy gradient is unnecessary and therefore the particle swarm optimization algorithm is particularly applicable to non-linear functions in Equation 1. We treat collision-free support vectors generated from the learning stage as the initial particles with an energy value. Given random searching velocities at the beginning of the optimization algorithm, the particles are updated in an iterative manner based on their velocities. The particle swarm optimization algorithm records the global best position among all the particles and the local best position for each particle. The particle's movement is influenced by its local best position and the global best position. As a result, every particle is expected to move towards its best position, w.r.t. the given energy function, and the sparsely distributed particles can fully explore configuration space. When particles explore the configuration space, the hand may collide with the object or have self-collisions between fingers. We use continuous collision detection [31] to compute the first instance of contact that avoids collisions.

When a feasible grasp pose is obtained, the force closure (FC) can be determined using the contacts between the fingers and the object. The grasp quality is then computed, which is related to the third term in Equation 1. If a grasp configuration is stable, we keep it in the grasp poses. As a result, we obtain a set of discrete stable grasp poses.

B. Iterative Learning

Algorithm 1 Learning Grasp Poses

Inpput: A robotic hand $\mathcal{A}$, a complex object $\mathcal{B}$ and the number of iterations N

Output: Stable Grasp Poses G

```
while N > 0 do
    S ← sampling(A, B)
    Support Vectors V ← SVM(S)
    N = N − 1
end while
particles ← k-means(V)
G ← PSO(A, B, particles)
return G
```

Some regions in the high-dimensional configuration space are not explored at all and some local regions have not been examined. To address this issue, we use iterative learning to add more new samples into the training data. Some samples are selected randomly in order to explore unknown regions and some are chosen near existing support vectors. Algorithm 1 gives the pseudocode of interactive learning grasp poses.

We sampled $5,000$ training samples in each iteration as the input of SVM to explore configuration spaces. Table I shows the time of the learning process and the number of trained collision-free support vectors. We chose collision-free support vectors in SVM as particles during the particle swarm optimization process. However, the number of collision-free support vectors in SVM is large and different for each object, computing particle swarm optimization algorithm with all collision-free support vectors is time-consuming. We used k-means clustering to compute 200 representative

TABLE I

MODEL COMPLEXITY AND LEARNING RESULTS OBTAINED FROM SVM

Model	tris	genus	Iterative Learning on CPU (s)					# of collision-free support vectors
			1	2	3	4	total	
mug	1746	1	9.4	40.2	127.3	279.3	456.2	3198
valve wheel	2388	5	11.4	42.1	126.6	284.5	464.6	2983
knot	7696	3	11.5	44.3	128.1	290.0	474.0	3039
6-genus knot	2420	6	11.1	42.6	124.8	281.0	459.5	2996
Olympic logo	2880	9	12.2	45.09	135.81	308.3	501.4	3191
baby teether	6956	12	11.3	42.9	127.1	287.0	468.3	3023
porous sphere	11516	>20	16.6	59.7	174.2	402.9	653.4	4400

support vectors as particles. Table II shows the best grasp quality estimated during the optimization process, the time of optimization, and the number of stable grasp poses found during the optimization. Some of the grasps shown in Fig. 4. We also computed force closure and grasp quality for each stable grasp pose. Table II shows the average quality for all stable grasp poses.

C. Time Complexity

The time spent in the learning stage depends on the number of samples, the time of collision detection, and the time to learn an approximation of configuration space. The collision detection is accelerated using precomputed bounding volume hierarchies of finger links and objects. The time spent in the stable grasp computation stage depends on the number of particles and the number of iterations. In each iteration, we perform collision detection for each hand configuration and compute its energy function value using Equation 1.

V. INTERACTIVE GRASPING FOR HIGH-GENUS OBJECTS

In this section, we present our benchmark models and the performance of our algorithm on various complex objects.

A. Interactive Grasping

At runtime, for a given robotic hand posture, we use kNN to search for the nearest stable grasp. In practice, our interactive algorithm can perform interactive grasping operations in less than 20ms for complex objects, even for high-genus objects with many holes.

B. Performance Evaluation

We evaluated the performance of our algorithm on a few high-genus complex objects, shown in Fig. 3. Table I shows their complexity and the number of genus. We use a three-fingered Barrett hand (see Figure 4) and a multi-fingered robotic hand (see Figure 1) to grasp these objects. We implement our learning grasp space algorithm using the GraspIt! library [23]. The performance is measured using a single thread on a PC with a 3.4GHz Intel i7 CPU and 8GB memory. When learning the grasp poses for each object during pre-processing, we set $\omega_1 = 0.02$ and $\omega_2 = 1.0$, as suggested in [6]. In addition, we set $\omega_3 = 20$.

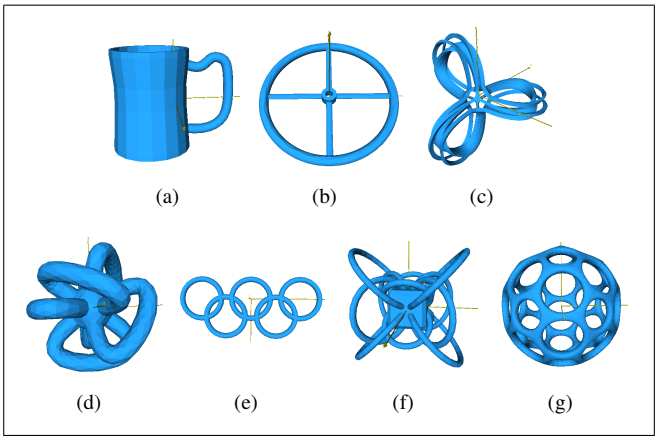


Fig. 3. Complex objects used in our experiments. (a) Mug; (b) Valve wheel; (c) Knot; (d) 6-genus knot; (e) Olympic logo; (f) Baby Teether; (g) Porous sphere.

VI. CONCLUSIONS

In this paper, we considered the grasping problem using a multi-fingered robotic hand under the assumption that the 3D models of the objects are known and that the shape of the object is complex. We presented a new algorithm to learning grasp poses for high-genus complex objects. Our experiment shows that our algorithm works for a wide range of complex objects and multi-fingered hands.

Our algorithm was tested in simulation and it has some limitations. First, our algorithm assumes the object-space representation of a complex object is available. Second, it is limited to rigid virtual objects and does not consider runtime deformation (e.g. twirling an object). In our future work, we are interested in extending our algorithm to handle partially-known objects and deformable virtual objects. Moreover, we would like to improve the performance using hardware acceleration so that real-time performance can be achieved.

ACKNOWLEDGMENT

This research work was supported by the NSFC (No.61631166002, No.61572196, No.61532002, No.61672237), National High-tech R&D Program of China (863 Program) under Grant 2015AA016404. Hao Tian was supported by the China Scholarship Council.

TABLE II

THE RESULTS DURING OPTIMIZATION STAGE.

Model	Best Grasp Quality during Optimization					time (s)	stable grasps	average grasp quality
	100	200	300	400	500			
mug	0.155	0.304	0.304	0.311	0.311	740.8	419	0.102
valve wheel	0.032	0.033	0.033	0.108	0.108	792.3	374	0.014
knot	0.114	0.127	0.16	0.16	0.2	899.1	339	0.08
6-genus knot	0.078	0.164	0.213	0.267	0.389	780.6	281	0.05
Olympic logo	0.022	0.025	0.032	0.032	0.032	917.7	95	0.006
baby teether	0.104	0.164	0.181	0.299	0.299	909.5	484	0.082
porous sphere	0.191	0.2	0.2	0.216	0.216	1017.3	383	0.071

REFERENCES

[1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 424–432, 2003.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[3] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *IEEE International Conference on Robotics and Automation*, vol. 348, p. 353, 2000.

[4] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis-A survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014.

[5] M. Ciocarlie, C. Goldfeder, and P. Allen. Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In *Robotics: Science and Systems Manipulation Workshop*, 2007.

[6] M. T. Ciocarlie. *Low-dimensional robotic grasping: Eigengrasp subspaces and optimized underactuation*. PhD thesis, Computer Science Department, Columbia University, 2010.

[7] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(2):58–73, 2002.

[8] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. Ph.d. dissertation, Carnegie Mellon University, Robotics Institute, 2010.

[9] S. El-Khoury and A. Sahbani. Handling objects by their handles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, number EPFL-TALK-168926, 2008.

[10] S. Elkhoury and A. Sahbani. On computing robust n-finger force-closure grasps of 3D objects. *International Conference on Robotics and Automation*, pp. 2480–2486, 2009.

[11] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pp. 2290–2295, 1992.

[12] M. Fischer and G. Hirzinger. Fast planning of precision grasps for 3D objects. *Intelligent Robots and Systems*, 1:120–126, 1997.

[13] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof. Grasp planning via decomposition trees. In *IEEE International Conference on Robotics and Automation*, pp. 4679–4684, 2007.

[14] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *IEEE International Conference on Robotics and Automation*, pp. 1710–1716, 2009.

[15] B. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard. Learning a real time grasping strategy. In *IEEE International Conference on Robotics and Automation*, pp. 593–600, 2013.

[16] K. Huebner and D. Kragic. Selection of robot pre-grasps using box-based shape approximation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1765–1770, 2008.

[17] P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Transactions on Graphics*, 25(3):872–880, 2006.

[18] J.-W. Li, H. Liu, and H.-G. Cai. On computing three-finger force-closure grasps of 2-D and 3-D objects. *IEEE Transactions on Robotics and Automation*, 19(1):155–161, 2003.

[19] Y. Li, J. L. Fu, and N. S. Pollard. Data-driven grasp synthesis using shape matching and task-based pruning. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):732–747, 2007.

[20] C. K. Liu. Dextrous manipulation from a grasping pose. *ACM Transactions on Graphics*, 28(3):article 59, 2009.

[21] X. Markenscoff and C. H. Papadimitriou. Optimum grip of a polygon. *International Journal of Robotics Research*, 8(2):17–29, 1989.

[22] A. T. Miller and P. K. Allen. Examples of 3D grasp quality computations. In *IEEE International Conference on Robotics and Automation*, pp. 1240–1246, 1999.

[23] A. T. Miller and P. K. Allen. GraspIt! A versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[24] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, pp. 1824–1829, 2003.

[25] V.-D. Nguyen. Constructing force-closure grasps. *International Journal of Robotics Research*, 7(3):3–16, 1988.

[26] R. Pelossof, A. T. Miller, P. K. Allen, and T. Jebara. An SVM learning approach to robotic grasping. *IEEE International Conference on Robotics and Automation*, 4:3512–3518, 2004.

[27] F. T. Pokorny and D. Kragic. Classical grasp quality evaluation: New algorithms and theory. In *IEEE/RSJ Intelligent Robots and Systems*, 2013.

[28] F. T. Pokorny, J. A. Stork, D. Kragic, et al. Grasping objects with holes: A topological approach. In *IEEE International Conference on Robotics and Automation*, pp. 1100–1107, 2013.

[29] N. S. Pollard and V. B. Zordan. Physically based grasping control from example. In *Symposium on Computer Animation*, pp. 311–318, 2005.

[30] 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 International Conference on Intelligent Robots and Systems*, pp. 1781–1788, 2011.

[31] S. Redon, M. C. Lin, D. Manocha, and Y. J. Kim. Fast continuous collision detection for articulated models. *Journal of Computing and Information Science in Engineering*, 5(2):126–137, 2005.

[32] H. Rijpkema and M. Girard. Computer animation of knowledge-based human grasping. In *ACM SIGGRAPH*, pp. 339–348, 1991.

[33] A. Sahbani, S. El-Khoury, and P. Bidaud. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–336, 2012.

[34] J. Seo, S. Kim, and V. Kumar. Planar, bimanual, whole-arm grasping. In *IEEE International Conference onRobotics and Automation*, pp. 3271–3277, 2012.

[35] K. B. Shimoga. Robot grasp synthesis algorithms: A survey. *International Journal of Robotics Research*, 15(3):230–266, 1996.

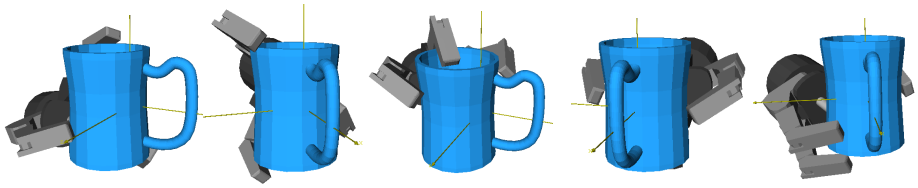
[36] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3):63, 2009.

[37] Y. Ye and C. K. Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics*, 31(4):41, 2012.

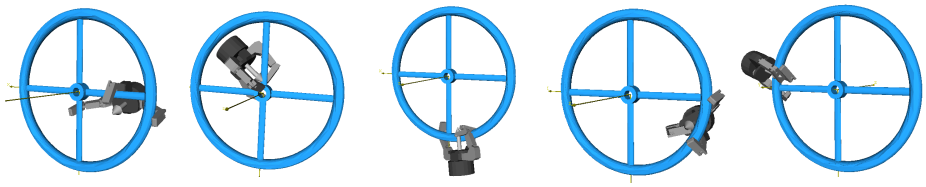
[38] F. Zacharias, C. Borst, and G. Hirzinger. Object-specific grasp maps for use in planning manipulation actions. *Advances in Robotics Research*, pp. 203–214, 2009.

[39] W. Zhao, J. Zhang, J. Min, and J. Chai. Robust realtime physics-based motion control for human grasping. *ACM Transactions on Graphics*, 32(6):207, 2013.

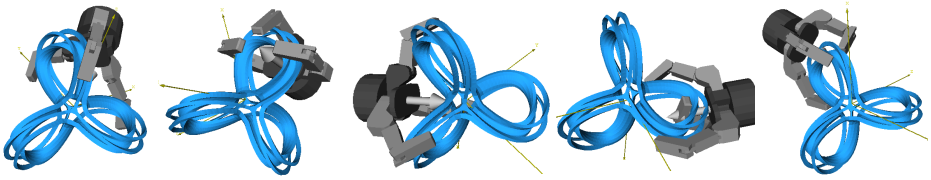
[40] Y. Zheng. An efficient algorithm for a grasp quality measure. *IEEE Transactions on Robotics*, 29(2):579–585, 2013.



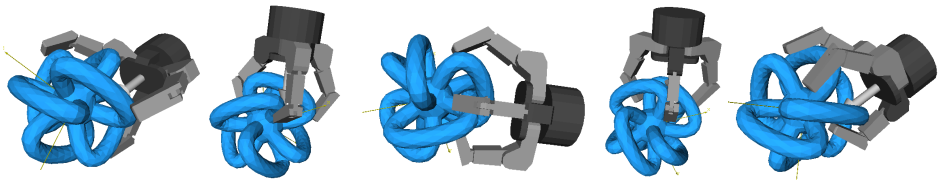
(a) mug



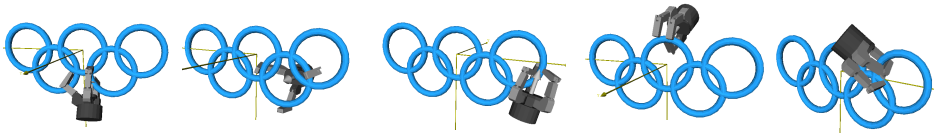
(b) vale wheel



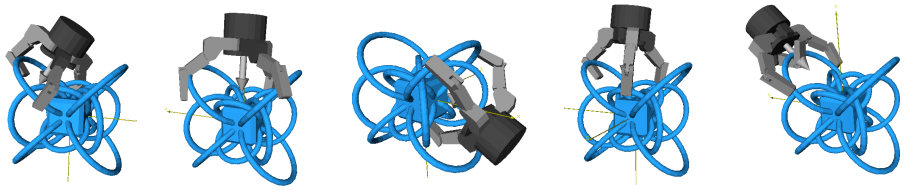
(c) knot



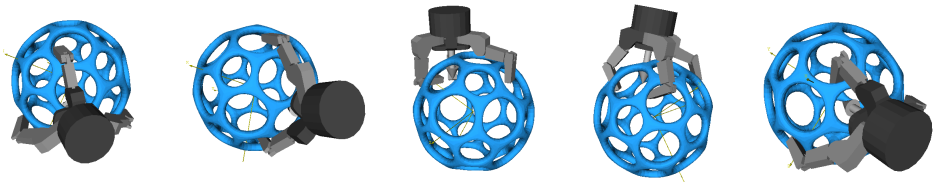
(d) 6-genus knot



(e) Olympic logo



(f) baby teether



(g) porous sphere

Fig. 4. Learning grasp poses for manipulating a wide range of complex objects.