Planning High-Quality Grasps Using Mean Curvature Object Skeletons

Nikolaus Vahrenkamp, Eduard Koch, Mirko Wächter, and Tamim Asfour

***Abstract*—In this letter, we present a grasp planner that integrates two sources of information to generate robust grasps for a robotic hand. First, the topological information of the object model is incorporated by building the mean curvature skeleton and segmenting the object accordingly in order to identify object regions which are suitable for grasping. We show how this information can be used to derive different grasping strategies, which also allows to distinguish between precision and power grasps. Second, the local surface structure is investigated to construct feasible and robust grasping poses by aligning the hand according to the local object shape. We apply the approach to a wide variety of object models of the KIT and the YCB real-world object model databases and evaluate it with several robotic hands. The results show that the skeleton-based grasp planner is capable to generate high-quality grasps in an efficient manner. In addition, we evaluate how robust the planned grasps are against hand positioning errors as they occur in real-world applications due to perception and actuation inaccuracies. The evaluation shows that the majority of the generated grasps are of high quality since they can be successfully applied even when the hand is not exactly positioned.**

***Index Terms*—Grasping, multifingered hands.**

I. Introduction

THE ability to robustly plan feasible grasps for arbitrary objects is an important capability of autonomous robots in the context of mobile manipulation. It allows to generate grasping configurations in an autonomous and unsupervised manner based on the object's 3D mesh information. Part-based grasp planning is an approach to deal with the complexity of arbitrary shaped objects by segmenting them into simpler parts which are more suitable for shape analysis for grasp generation. This procedure complies also with the behavior of humans which prefer to structure objects into smaller segments (see e.g., the recognition-by-components paradigm of Biederman [1]). Representing an object by its parts fits very well to robotic grasp planning, since a robotic gripper or hand can usually only grasp an object by accessing a local surface area for establishing contacts.

Manuscript received September 10, 2017; accepted December 14, 2017. Date of publication January 12, 2018; date of current version February 1, 2018. This letter was recommended for publication by Associate Editor Y. Zheng and Editor H. Ding upon evaluation of the reviewers' comments.This work was supported by the European Unions Horizon 2020 Research and Innovation programme under Grant 643950 (SecondHands). *(Corresponding author: Nikolaus Vahrenkamp.)*

The authors are with the High Performance Humanoid Technologies (H^2T) Laboratory, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany (e-mail: vahrenkamp@kit.edu; eduard.koch@hotmail.de; mirko.waechter@kit.edu; asfour@kit.edu).

Digital Object Identifier 10.1109/LRA.2018.2792694

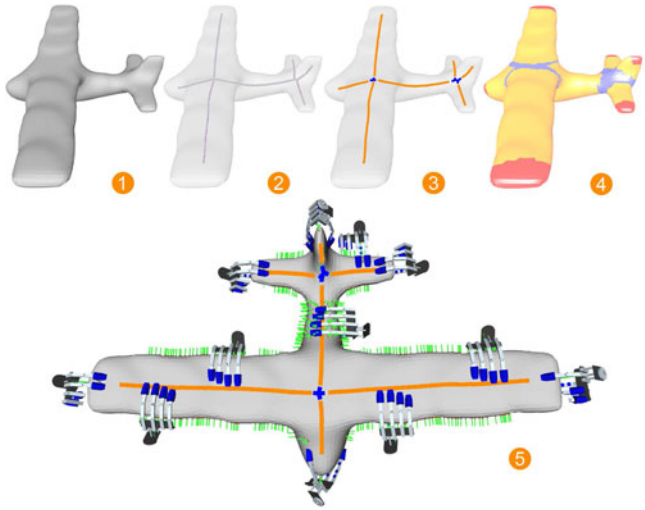

Fig. 1. The object mesh model (1) is used to build the mean curvature object skeleton (2). The skeleton is segmented (3) and the corresponding surface segments (4) are depicted in red (segment end points), yellow (connecting segments), and blue (segment branches). The results of the skeleton-based grasp planner are visualized (5) by indicating the approach directions of the planned grasps via green lines. In addition, a selection of grasps is fully visualized.

The part-based grasp planner presented in this work is based on the assumption that an object can be segmented according to its skeleton structure. The object skeleton simplifies the topological representation and provides information about the connectivity of the segments. Object segmentation is realized by analyzing the skeleton according to branches and crossings in the topology. The object segments are analyzed separately to determine if several grasping strategies (e.g., power or precision grasps) can be applied according to the local surface information (see Fig. 1). Since robustness is essential for robotic grasp planning, it is desirable to plan grasps which can be executed robustly in realistic setups. Hence, we aim at planning grasps that are robust to disturbances and inaccuracies as they appear during execution due to noise in perception and actuation. In this work, we show that the analysis of the object skeleton in combination with the usage of local surface properties leads to grasping hypotheses which are robust to inaccuracies in robot hand positioning, which indicates a reliable execution on real robot platforms.

The proposed skeleton-based grasp planning approach makes use of an analysis of the object structure to identify object regions on which a high probability exists that grasps can be applied. In addition, we show that an analysis of the local object

surface, encoded by grasping strategies, leads to grasping hypotheses with high force-closure and robustness rates. An implementation of the presented grasp planning approach based on the Simox library [2] is provided as a C++ open source project.[1]

The paper is organized as follows. Section II introduces related work while Section III describes the mean curvature object skeleton, mesh segmentation and local object properties analysis. Section IV describes the part-based grasp planning and Section V presents the results. Section VI conclude the contribution of the paper.

II. Related Work

Napier divided hand movements in humans into *prehensile movements* and *non-prehensile movements* to distinguish between suitable and non-suitable grasping movements [3]. He showed that prehensile grasping movements of the hand consist of two basic patterns which he termed precision grasp and power grasp. Based on this work Cutkosky developed a grasp taxonomy which distinguishes between 16 different grasping types [4]. The transfer of grasping strategies in humans to robotic applications is usually done by reducing the complexity, e.g., by considering a low number of grasp types (i.e., power and precision grasp) or by using the Eigengrasp approach which is capable of approximating human grasping movements with low degree of freedom spaces [5].

Approaches for grasp synthesis in the literature are usually divided into analytical and empirical or data-driven algorithms [6], [7]. Analytical approaches are based on geometrical properties and/or kinematic or dynamic formulations, whereas data driven approaches rely on simulation. The complexity of the problem is often reduced by utilizing randomized methods and by considering simplified contact models. Data-driven approaches also make use of simulation environments such as *GraspIt!* [8], OpenRave [9], and Simox [2] to generate and evaluate grasping poses. Many works generate grasping hypotheses and evaluate them by applying the Grasp Wrench Space approach which allows to determine if a grasp is *force-closure* (i.e., valid) and to compute ϵ or *Volume* quality values indicating how much the grasping configuration can resist to external forces [10], [11].

The execution of such grasps with real robot hands is challenging since small disturbances, as they appear during real-world execution, could quickly lead to unstable grasping configurations as shown in [12]. It has been shown that such potential inaccuracies in execution can be considered during grasp planning by analyzing small variations of the generated grasping pose.

In part-based grasp planning there exist several approaches which consider parts of the object for grasp synthesis. In [13] the object is approximated by a set of primitive shapes (box, cylinder, sphere, cone) to which manually defined hand configurations can be assigned. A similar approach has been presented in [14]. The object is decomposed in superquadrics and several heuristics are used to generate the grasping information. In [15] the object is decomposed based on Minimal Volume Bounding Boxes. grasping information is synthesised on the resulting bounding boxes. The grasp planner presented in [16] is operating on a topological object segmentation which is computed based on the Reeb Graph formalism. The resulting object segments are used for randomized grasp generation. Independent Contact Regions can be used to determine surface areas that are suitable for grasping [17]. An voxelized object representation is used by [18] to plan feasible grasps via hand-object geometric fitting strategies.

Our previous work described in [19], [20] is related to the work we present here since it uses the medial axis transform as a simplified object representation on which grasp planning is performed. The medial axis representation is computed on point clouds or mesh data and provides information about the object structure and object symmetries. Before grasp planning can be performed, the medial axis transform is processed via cluster algorithms and transferred to grid-based data structures on which grasp synthesis is realized based on several heuristics for typical grid structures. In contrast to [20], we use mean curvature skeletons [21] to represent the object structure. Compared to the medial axis approach, this representation results in a reduced complexity while preserving full object surface information (see Section III). We exploit the skeleton structure for object segmentation and in Section IV we show how the object segments can be analyzed for part-based grasp planning. The approach is evaluated in Section V on several object data sets by investigating the force closure rate and the robustness of the generated grasps according to [12]. In addition, we compare the results to a randomized grasp planning algorithm which aligns the approach direction to the object surface [2], similar to the approach used in [9] and [22].

III. Object Skeletons

3D mesh objects are processed in order to generate mean curvature skeletons which provide a medically centered skeleton representing the object's topology [21]. As we show in this section, the skeleton data structure is used to segment the object according its topology.

A. Mean Curvature Skeletons

Mean curvature skeletons are generated by a contraction-based curve skeleton extraction approach. As input, the 3D mesh of the object is processed in order to retrieve a regularized surface which results in a triangulated object data structure. The set of object surface points are denoted by $O = o_0, \ldots, o_{n-1}$. As described in [21], the skeleton is build based on iterative mesh contraction via a mean curvature flow approach. Several results are depicted in Fig. 2.

A resulting skeleton is a graph $S = (V, E)$, in which each vertex $v \in V$ is connected to one or multiple neighbors via edges $e \in E$. A vertex $v_i = \{s_i, P_i\}$ consist of the 3D position of the skeleton point $s_i \in R^3$ and a set of uniquely associated points on the object surface $P_i \subset O$. This is an important property of the mean curvature flow skeleton algorithm which is used later when we analyze the local object shape of a skeleton interval on

[1] https://gitlab.com/Simox/simox-cgal

Fig. 2. The mean curvature skeleton of several objects.

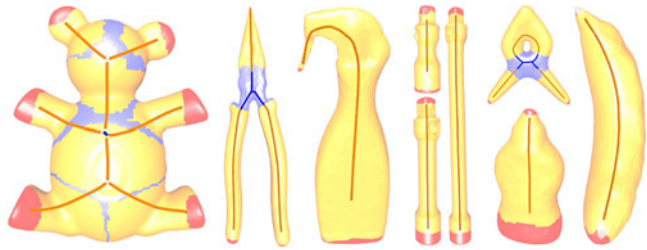

Fig. 3. The segmentation of several objects together with the computed skeletons. The surface is colored according to the corresponding skeleton. Branching areas are colored in blue, endpoints result in red color, and the surface associated with connecting segments is visualized in yellow.

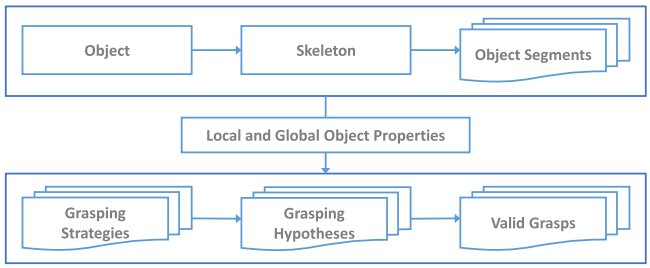

Fig. 4. The grasp planning process.

basis of the corresponding object surface points. More details on the mean curvature flow skeleton approach can be found in [21]. Since all surface points of the object are uniquely associated with a skeleton point, the following relation holds $\sum |P_i| = |O|$. An edge $e = \{v_a, v_b\}$ connects the two vertices v_a and v_b.

B. Mesh Segmentation

The object is segmented based on the skeleton structure in order to generate segments which represent the object's topology. For further processing, each skeleton vertex v is classified according to its connection to neighboring vertices:

Branching Vertex: Such vertices represent branches or crossings in the skeleton structure. As expressed in (1), a vertex v is a branching vertex if there exist more than two edges in the skeleton $S = (V, E)$ containing v.

$$|\{e \in E : v \in e\}| > 2 \Leftrightarrow \text{v is a branching vertex} \quad (1)$$

Endpoint Vertex: An endpoint vertex v is connected to exactly one other vertex (see (2)).

$$|\{e \in E : v \in e\}| = 1 \Leftrightarrow \text{v is an endpoint vertex} \quad (2)$$

Connecting Vertex: A connecting vertex v connects to exactly two neighbors as expressed in (3).

$$|\{e \in E : v \in e\}| = 2 \Leftrightarrow \text{v is an connecting vertex} \quad (3)$$

The mesh can now be easily segmented by analyzing the skeleton structure and grouping the skeleton vertices according to their connections. A segment $S_i \subset S$ is defined as follows:

$$\begin{aligned} &\forall v \in S_i : \text{v is an connecting vertex} \wedge \\ &\forall e = \{v_a, v_b\} \in S_i : v_a, v_b \in S_i. \end{aligned} \quad (4)$$

The resulting segments S_i, contain sub graphs of S consisting of connecting vertices which are enclosed by branching or endpoint vertices.

As depicted by several exemplary results in Fig. 3, the object segmentation provides valuable information about the object structure which can be exploited for grasp planning as we show in the following section.

IV. Part-based Grasp Planning

The object skeleton and the resulting segmentation can now be used to analyze global object properties based on structural and topological information. This information is e.g., used to prefer specific topological structures for grasp planning or to identify possible approach directions. In addition, we compute and analyze local object properties as a basis for the decision if a specific grasping strategy can be applied. These local object properties are related to a specific skeleton point and provide several parameters which describe the local object surface allowing to simplify the process of local shape analysis. Our approach allows for defining different grasping strategies based on global and local object information. Although, the approach is not limited to a fixed set of grasping strategies, we investigate the most common grasping strategies in the context of robotic grasping which relate to power and precision grasps. Fig. 4 depicts an overview of the grasp planning process.

The grasp planner is shown in Algorithm 1. First, the method *nextSkeletonVertex* determines the next skeleton vertex v_i. To do so, the whole skeleton is uniformly sampled according to the skeleton vertex distance parameter d, which defines the minim distance between two consecutive skeleton vertices. Then, we iterate through all defined grasping strategies and calculate several local object properties. The properties P are used to evaluate if the grasping strategy gs is applicable to the vertex v_i. If so, several grasping hypotheses are created. For each hypothesis h, a validity check is applied to ensure the correctness of the result. All valid hypotheses are finally stored and returned.

A. Local Object Properties

To verify that a grasping strategy is applicable, we define several local object properties which can be derived from a skeleton vertex $v = (s, P)$:

Vertex Type P_T: Identifies the type of vertex $P_T \in \{connection, endpoint, branch\}$.

Grasping Interval P_I: Starting from v, P_I includes all outgoing skeleton sub graphs until either a branching or endpoint

Algorithm 1: Grasp Planner.

Input: skeleton S, grasping strategies GS, vertex dist. d
Output: set of valid grasps G

$G = \emptyset$
while (!timeout() $\wedge$ verticesAvailable()) **do**
  v_i = nextSkeletonVertex(S, d)
  forall ($gs \in GS$) **do**
    P = calculateLocalObjectProperties(v_i, gs, S)
    if (evaluateGraspingStrategy(gs, P)) **then**
      H = generateGraspingHypotheses(v_i, gs, P)
      forall ($h \in H$) **do**
        if (isValid(h)) **then**
          $G = G \bigcup \{h\}$
return G

vertex is reached or a maximum length is travelled on the corresponding graph. Hence, $P_I = \{SG_0, \ldots, SG_s\}$ contains the sub graphs $SG \subset S$ starting from v resulting in $|P_I| = 1$ for endpoint vertices, $|P_I| = 2$ for connecting vertices and $|P_I| > 2$ for branching vertices. Depending on the evaluated strategy, the maximum path length is set to half of the robot hand's width (power grasps) or to the width of one finger (precision grasp). This information is of interest to determine if a connection segment offers enough space for applying a grasp. In addition, the curvature κ of each point on the grasping interval needs to be limited in order to avoid sharp edges which are difficult to capture in terms of aligning the hand for grasp planning. Hence, a sub graph is cut if the curvature at a specific skeleton point is too high. The curvature at a skeleton point s is defined as

$$\kappa(s) = \frac{||s' \times s''||}{||s'||^3},$$

with the first and second derivatives s' and s'' which can be derived by numerical differentiation [23].

Local Surface Shape P_{SS}: P_{SS} describes the local surface of the object by providing several parameters. First, we reduce the complexity of the analysis of the local object shape by introducing a grasping plane onto which the associated surface points of P_I are projected. The plane is defined by the skeleton point s and the normal parallel to the tangent in s. The considered surface points cover all associated surface points of the skeleton vertices in P_I (see Fig. 5). This is possible since the grasping interval P_I contains a set of skeleton vertices and for each skeleton vertex there exists a unique association to object surface points.

The projected surface points are analyzed by applying a principal component analysis to determine the eigenvalues λ_1, λ_2 and the eigenvectors ev_1, ev_2. In the following, λ_2 is used to identify the local thickness of the object. For further processing, the ratio $r = \frac{\lambda_1}{\lambda_2}$ is calculated and a threshold t_r is used to distinguish between round and rectangular surface shapes. Throughout this work, we use $t_r = 1.2$.

$$shape = \begin{cases} round & \text{if } r < t_r \\ rectangular & \text{otherwise} \end{cases} \quad (5)$$

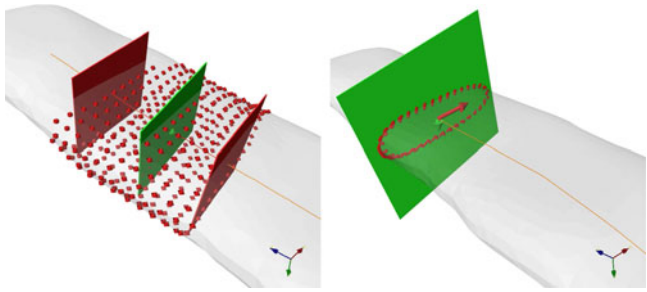

Fig. 5. Left: Based on a skeleton vertex v (shown in green), a grasping interval P_I is depicted. The red planes define the borders of P_I and the red points visualize the object surface points which are associated with P_I. On the right, the grasping plane is depicted in green and the projected surface points are shown in red. The corresponding eigenvectors of the projected surface points are visualized as red and green arrows.

Finally, the local surface shape is represented through the 5-tuple $P_{SS} = (\lambda_1, \lambda_2, ev_1, ev_2, shape)$. Note, that with this analysis of the local object shape, we again introduce a simplification in order to catch the most common shapes of object parts. By introducing a more sophisticated analysis of the local object shape, the whole approach could be extended and improved in a way that more complex object parts could be processed.

B. Grasping Strategies

Our approach allows for setting up a variety of grasping strategies based on the available local and global object information. The grasping strategies can be interpreted as heuristics, which encode promising strategies according to the available hand and object parameters. In the following, we describe several grasping strategies which can be used to generate *precision* and *power* grasps on connection and endpoint parts of the skeleton. To evaluate if a grasping strategy gs can be applied, the local object properties are analyzed as described in Table I. Note, that with the following grasping strategies we only cover several straightforward approaches to apply power or precision grasps. One could easily define more sophisticated grasping strategies by performing a more detailed analysis of local and global object properties.

The grasping strategies can be interpreted as follows:

1) Precision Grasp on Connecting Segments: This strategy is applied on a vertex of a connection segment, which means that exactly two skeleton intervals are available in P_I. The length of each interval has to be at least $fingerwidth$ resulting in an accumulated length of the local object skeleton intervals of two times the width of an finger which is reasonable for applying precision grasps. In addition, we distinguish between *round* and *rectangular* shapes of the local object surface. For *round* shaped objects, we evaluate if the local object thickness, identified by λ_2 is within the range $[pre_2^-, pre_2^+]$. In case the shape is *rectangular*, we additionally check if the local object length λ_1 is within $[pre_1^-, pre_1^+]$ in order to bias the decision towards power grasps on objects which provide a reasonable depth.

2) Power Grasp on Connecting Segments: Similar to the precision grasp strategy, we analyze the length of both skeleton

TABLE I
GRASPING STRATEGIES ARE DEFINED ACCORDING TO SEVERAL LOCAL OBJECT PROPERTIES

Grasping Strategy	Nr.	$\mathbf{P_T}$	Interval Length in $\mathbf{P_I}$	Shape	λ_1	λ_2
Precision Grasp on Connecting Segments	1a	con.	$\geq fingerwidth$	round	n/a	$[pre_2^-, pre_2^+]$
	1b	con.	$\geq fingerwidth$	rect.	$[pre_1^-, pre_1^+]$	$[pre_2^-, pre_2^+]$
Power Grasp on Connecting Segments	2a	con.	$\geq 0.5 \cdot handwidth$	round	n/a	$[pow_2^-, pow_2^+]$
	2b	con.	$\geq 0.5 \cdot handwidth$	rect.	$> pow_1^-$	$[pow_2^-, pow_2^+]$
Precision Grasp on Endpoint Vertices	3	endpt.	n/a	round, rect.	n/a	$[pre_2^-, pre_2^+]$
Power Grasp on Endpoint Vertices	4	endpt.	n/a	round, rect.	n/a	$[pow_2^-, pow_2^+]$

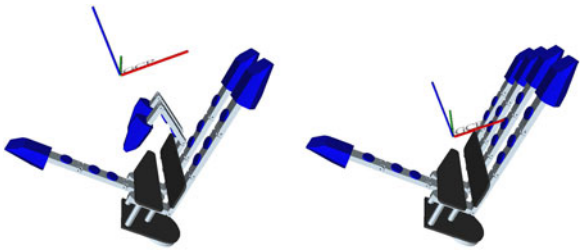

Fig. 6. The grasp center points of the ARMAR-III hand for precision and power grasps. The coordinate axes (x: red, y: green, z: blue) are used to align the hand.

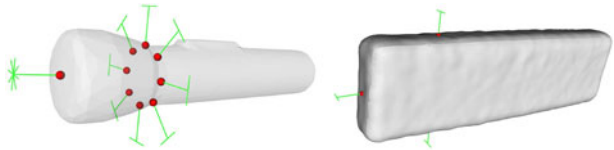

Fig. 7. The generated approach directions are depicted for an endpoint with local surface properties *round* (left) and *rectangular* (right). In addition, the approach directions for one skeleton point on a connecting segment are depicted for *round* and *rectangular* local surface properties.

intervals in P_I for a given vertex of a connection segment. The length of each interval has to be at least $0.5 \cdot handwidth$ in order to be able to apply a power grasp. In addition, we distinguish between *round* and *rectangular* shapes of the local object surface. For *round* shaped objects, we evaluate if the local object thickness, identified by λ_2, is within the range $[pow_2^-, pow_2^+]$. In case the shape is *rectangular*, we may want to exclude small objects and therefore we additionally check if the local object length λ_1 is larger than pow_1^-. Since for rectangular shapes, the approach directions are only derived considering the thinner side (which is covered by λ_2), we do not need to specify an upper bound for λ_1 (see also Fig. 7 on the right).

3) Precision Grasp on Endpoint Vertices: This strategy is applied on endpoints of the skeleton structure. Similar to the grasping strategies on connecting segments, the local object shape is interpreted based on the properties of the grasping plane. The length of the local object shape has to be within the range $[pre_2^-, pre_2^+]$ in order to be able to apply a precision grasp.

4) Power Grasp on Endpoint Vertices: Power grasps are applied on endpoints if the local object length is within $[pow_2^-, pow_2^+]$.

C. Grasping Hypotheses

From each grasping strategy several grasping hypotheses are derived and evaluated for correctness (collision-free and force closure) in order to validate the generated grasp.

Grasp Center Points: For each hand model, we define a grasp center point (GCP) for precision and power grasps, identifying the grasping center point and the approaching direction [24]. The GCP_{pre} and GCP_{pow} for the robotic hand of ARMAR-III is depicted in Fig. 6. The approach direction is the z-axis of the coordinate system (shown in blue).

Building Grasping Hypotheses: For a given skeleton vertex v, all valid grasping strategies are evaluated and a set of grasping hypotheses is derived. Therefore, a set of potential approach directions and corresponding hand poses is determined as follows.

Hand Orientation: The *shape* entry of the Local Surface Shape property P_{SS} results in the generation of different approach directions. In case of a *round* local surface, the approach direction is uniformly sampled around the skeleton point s. In this case, the approach directions are perpendicular to the skeleton tangent in s for connecting segments and aligned with the skeleton tangent for endpoints. To fully specify an approach direction three parameters are needed. In case of connection segments these parameters are: a) the z-axis of GCP is perpendicular to the skeleton tangent; b) the y-axis of the GCP is parallel to the tangent; c) the remaining parameter is uniformly sampled. In case of endpoint segments, two parameters are specified by aligning the z-axis of the GCP to the tangent in s. The remaining parameter is again specified by uniform sampling. If the local object shape evaluates to *rectangular*, four approach directions are built to align the robot hand according to the eigenvectors ev_1 and ev_2. In Fig. 7, the generated approach directions for one endpoint and one skeleton point on a connection segment are depicted for *round* (left) and *rectangular* (right) local surface properties. In both figures, the approach direction is projected along the negative approach direction onto the object surface. It can be seen that a *round* local surface results in uniformly sampled orientations (in this example, there are eight directions generated for an endpoint, respectively 16 for a skeleton point on a connecting segment). The right figure shows how a *rectangular* local surface results in two approach directions for an endpoint and four approach directions for a skeleton point on a connection segment. Based on this set of hand orientations, which is computed for a specific skeleton vertex, the full grasping pose is computed in the next steps.

Hand Position: The initial position of the hand is derived from the skeleton point s. This position is extended to full 6D hand poses by combining it with all computed hand orientations of the preceding step.

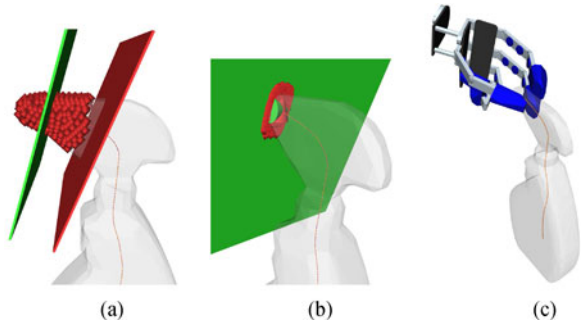

Fig. 8. (a) The grasping interval together with all associated surface points for a skeleton end point. (b) The grasping plane together with the projected surface points. (c) A generated grasp based on the grasping strategy 1b.

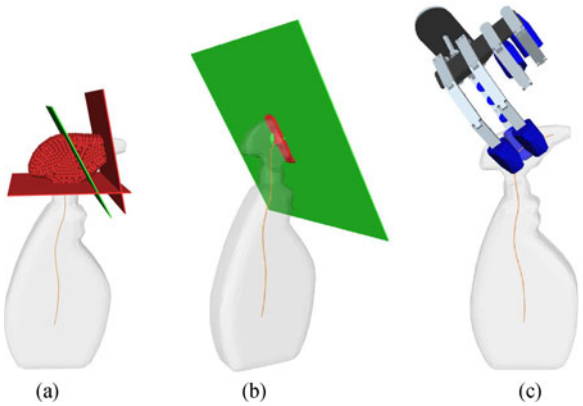

Fig. 9. (a) The grasping interval together with all associated surface points for a connection skeleton point. (b) The grasping plane and the projected surface points. (c) A generated grasp based on the grasping strategy 3.

Retreat Movement: To generate a valid grasping hypothesis, the hand model is moved backwards (according to the approach direction) until a collision-free pose is detected. This procedure is aborted if the movement exceeds a certain length.

In Figs. 8 and 9 the grasping interval, the grasping plane and a generated grasp are depicted for an endpoint vertex and a connection segment respectively.

D. Validation of Grasping Hypotheses

All generated grasping hypotheses are evaluated by closing the fingers, determine the contacts on the object model and by determining if the contacts result in a force-closure grasp. In addition, the quality of the grasp in terms of the grasp wrench space ϵ value is computed. For grasp stability analysis we employ the methods provided by the Simox library [2]. All force closure grasps are stored in the set G of valid grasps. In Fig. 10 the resulting set of grasps are depicted for several objects of the KIT object model database [25] and the Yale-CMU-Berkeley (YCB) Object and Model Set [26]. Note, that the object models were generated based on real-world point cloud data, i.e., no artificial shapes are used for grasp planning. The resulting grasps are visualized by projecting the GCP onto the surface according to the corresponding approach direction. The orientation of the grasp is visualized via a green bracket. A selection of the generated grasps for different hands is additionally shown in Fig. 11.

V. Evaluation

The skeleton-based grasp planning approach is evaluated by applying the algorithm on a wide variety of objects of the Yale-CMU-Berkeley (YCB) Object and Model Set [26] and the SecondHands subset of the KIT object model database [25]. All object models have been generated by processing point cloud scans of real-world objects. By using such realistic object models, we show that the grasp planning approach can be used under real-world conditions, i.e., without depending on perfectly modeled object meshes.

We compare the results of our approach with a randomized grasp planner [2] which generates power grasps by aligning the approach direction of the end effector to surface normals and evaluates the success by closing the fingers, determining the contacts and evaluating force closure and the grasp quality with the grasp wrench space approach. The planner generates similar results as the approach used in [9] and [22].

In total, we use 83 objects, 69 from the YCB data set (we exclude all objects with degenerated meshes) and 14 from the SecondHands subset of the KIT object database. The evaluation is performed with three different robot hand models: The ARMAR-III hand, the Schunk Dexterous Hand, and the Shadow Dexterous Hand (see Fig. 11). For all hands we define a precision and power preshape with corresponding GCP information. In total, we generated around 4000 valid grasps for each hand.

A. Efficiency and Force-Closure Rate

The first evaluation concerns the performance of the approach in terms of efficiency and force closure rate. In Table II, the results of the reference planning approach and the skeleton-based grasp planner are shown for the ARMAR-III hand, the Schunk Dexterous Hand, and the Shadow Dexterous Hand. The table shows the measured mean values together with the standard deviation over all grasps that were generated on all 83 considered objects. The time that is needed for generating a valid grasp directly depends on the complexity of the involved 3D models of the hand since many collision and distance calculations need to be performed when closing the fingers for contact detection. The comparison shows that the skeleton-based approach outperforms the surface-based planner in terms of efficiency (time needed to generate a grasping pose) and force-closure rate (number of force-closure grasps in relation to all generated grasps). Note, that both algorithms validate the computed grasping hypotheses and hence, only force-closure grasps are reported. With this evaluation, we show how good the grasping strategies apply to the different objects and that the vast majority of the grasping hypotheses result in force-closure grasps.

When looking at the resulting grasping poses in Fig. 11, it can be seen that the planned grasping poses are of high quality in terms of what a human would expect how a robot should grasp an object. Although we cannot provide any numbers on the *human-likeness* of the generated grasps, the underlying algorithm produces good grasps in this sense since the robot hand is aligned with the structure of the object's shape.

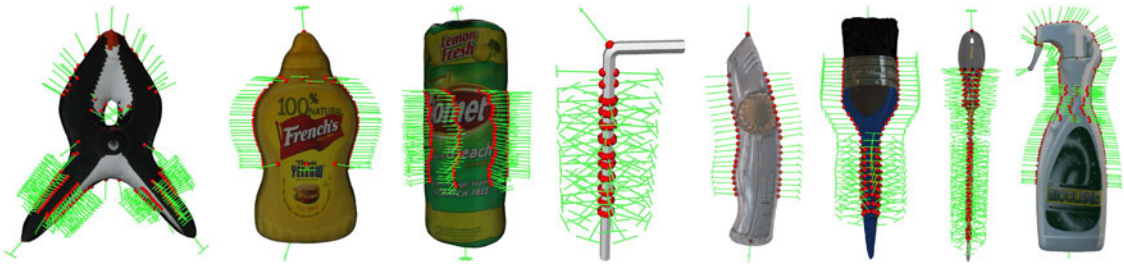

Fig. 10. Results of the skeleton based grasp planner with several objects of the KIT and YCB object databases. The red dots and the green lines depict the approach movements of the corresponding grasp.

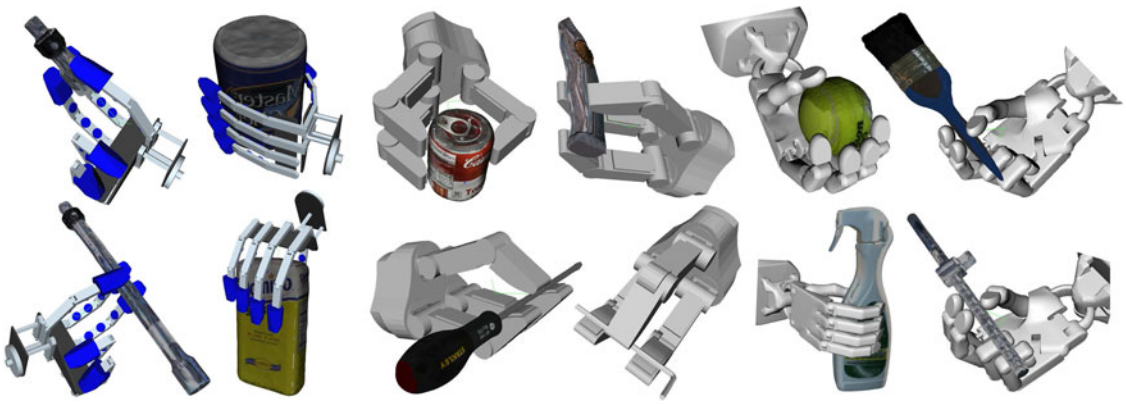

Fig. 11. Excerpt of the generated grasps for the three hand models that were used in the evaluation.

TABLE II
RESULTS OF THE EVALUATION

	Avg. Time per Grasp	Force-Closure Rate	Robustness Score r
Surface-based Grasp Planner			
ARMAR-III Hand	32.29 ms	57.80%	57.17%
	±23.87 ms	±28.54%	±16.18%
Schunk Dext. Hand	67.43 ms	70.78%	61.94%
	±23.22 ms	±21.72%	±27.39%
Shadow Dext. Hand	90.88 ms	43.55%	55.34%
	±34.05 ms	±28.32%	±18.83%
Skeleton-based Grasp Planner			
ARMAR-III Hand	12.19 ms	95.74%	94.23%
	±3.47 ms	±11.33%	±9.52%
Schunk Dext. Hand	12.98 ms	86.69%	94.87%
	±2.72 ms	±20.50%	±9.74%
Shadow Dext. Hand	28.68 ms	85.53%	76.46%
	±11.34 ms	±25.40%	±20.36%

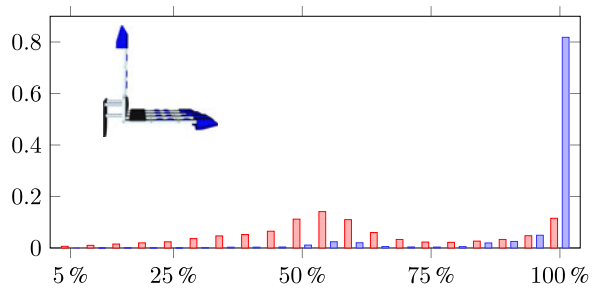

Fig. 12. Robustness histograms for the reference grasp planning approach (red) and the skeleton-based grasp planner (blue) with the ARMAR-III hand. The histogram bins are arranged on the x-axis, the y-axis indicates the absolute percentage.

B. Robustness

We evaluate the robustness of the generated grasping information by investigating how inaccuracies in hand positioning would affect the success rate of the grasping process. Related to the approach in [12], we compute a robustness score r for each grasp which indicates how many pose variances within a certain distribution result in a force-closure grasp. We create erroneous variances of the grasping pose p by applying a random position and orientation offset to the pose of the grasp. As proposed in [12], the offset is applied w.r.t. the center of all contacts. The resulting pose p' is then evaluated by moving the hand to p', closing the fingers, detecting the contacts and evaluating if the pose would result in a force-closure grasp. If p' results in an initial collision between hand and object, we count this pose as a failed sample, although there might be numerous grasps which could be executed. To get more detailed information, such situations could be further investigated by applying a physical simulation and considering the approach movement.

The robustness score r is then generated by determining the percentage of force-closure grasps of the total number of displaced samples. In all our experiments we draw 100 samples of displaced grasping poses with a normal distributed error (standard deviation: 10 mm and 5 degree).

In Fig. 12, a histogram of the robustness scores for all generated grasps on all considered objects for the hand of ARMAR-III is shown. The histogram bins on the x-axis cover 5% each and the y-axis indicates the absolute percentage of each histogram bin. The same histogram is shown in Fig. 13 and in Fig. 14 for the Schunk Dexterous Hand and the Shadow Dexterous Hand

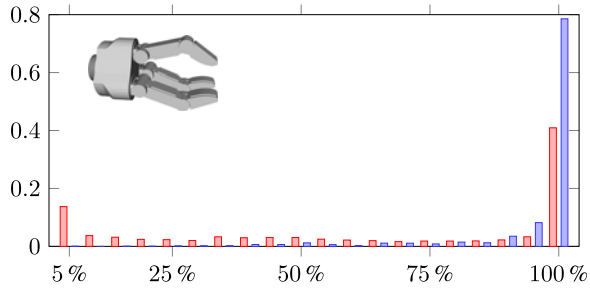

Fig. 13. Robustness histograms for the Schunk Dexterous Hand.

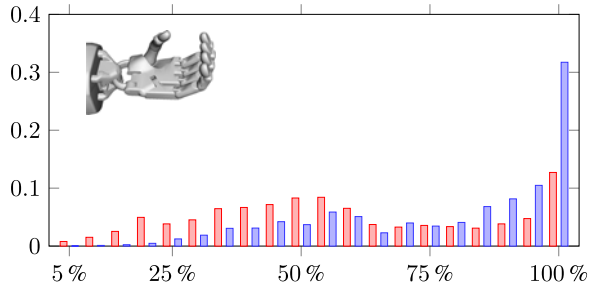

Fig. 14. Robustness histograms for the Shadow Dexterous Hand.

respectively. It can be seen, that the majority of the planned grasps of the skeleton-based approach are robust to disturbances.

As shown in Table II in the right column, the robustness score r could be considerably increased for all three hand models when using the skeleton-based grasp planning approach. It is above 94% for the ARMAR-III and the Schunk Dexterous Hand, which means that 94% of the investigated ill-positioned grasps were leading to force-closure configurations. The value for the Shadow Dexterous Hand is lower, which seems to be mainly caused by the fact that the hand kinematics is more complex and the corresponding preshapes provide less room for re-positioning. The execution of these high-quality grasps with a real robot manipulator would result in higher grasping performance since inaccuracies in perception and/or gripper positioning would still result in a successful grasp.

VI. Conclusion

In this work, we presented a grasp planning approach which takes into account global and local object properties to generate stable and robust grasping information for robotic hands. Global information is gathered by analyzing the object's mean curvature skeleton in order to identify suitable regions for applying a grasp. In addition, local information is used to select the grasp type and to align the hand according to local skeleton and surface properties. We showed that the approach is capable of generating high quality grasping information for real-world objects as they occur in object modeling databases such as the KIT or the YCB object DB projects. We evaluated the approach to a wide variety of objects with different robot hand models and showed that the resulting grasps are more robust in the presence of inaccuracies in grasp execution compared to grasp planners which do not consider local object properties. In addition, we think that if we compare the results of our grasp planner to the results of other approaches, the generated grasping poses look more natural, i.e., more human like, although we did not perform a qualitative evaluation in this sense.

References

[1] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychol. Rev.*, vol. 94, no. 2, pp. 115–147, 1987.
[2] N. Vahrenkamp *et al.*, "Simox: A robotics toolbox for simulation, motion and grasp planning," in *Proc. Int. Conf. Intell. Auton. Syst.*, 2012, pp. 585–594.
[3] J. R. Napier, "The prehensile movements of the human hand," *Bone Joint J.*, vol. 38, no. 4, pp. 902–913, 1956.
[4] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 269–279, Jun. 1989.
[5] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *Int. J. Robot. Res.*, vol. 28, no. 7, pp. 851–867, 2009.
[6] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis — A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.
[7] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robot. Auton. Syst.*, vol. 60, no. 3, pp. 326–336, 2012.
[8] A. T. Miller and A. T. Miller, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.
[9] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Aug. 2010.
[10] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. 1992 IEEE Int. Conf. Robot. Autom.*, 1992, pp. 2290–2295.
[11] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Auton. Robots*, vol. 38, no. 1, pp. 65–88, 2015.
[12] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *Proc. 2012 IEEE Int. Conf. Robot. Autom.*, 2012, pp. 557–562.
[13] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2003, vol. 2, pp. 1824–1829.
[14] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4679–4684.
[15] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1628–1633.
[16] J. Aleotti and S. Caselli, "A 3D shape segmentation approach for robot grasping by parts," *Robot. Autom. Syst.*, vol. 60, no. 3, pp. 358–366, 2012.
[17] M. A. Roa and R. Suarez, "Determination of independent contact regions on discretized 3D objects," in *Proc. 2007 IEEE Int. Symp. Assem. Manuf.*, Jul. 2007, pp. 191–196.
[18] P. Song, Z. Fu, and L. Liu, "Grasp planning via hand-object geometric fitting," *Visual Comput.*, vol. 1, pp. 1–14, 2016.
[19] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of balls for shape approximation in robot grasping," in *Proc. 2010 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1592–1599.
[20] M. Przybylski, M. Wächter, T. Asfour, and R. Dillmann, "A skeleton-based approach to grasp known objects with a humanoid robot," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2012, pp. 376–383.
[21] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," *J.Comput. Graph. Forum*, vol. 31, no. 5, pp. 1735–1744, 2012.
[22] D. Kappler, B. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 4304–4311.
[23] J. CASEY, *Exploring Curvature* (Vieweg Mathematics Series). New York, NY, USA: Springer, 1996.
[24] T. Asfour *et al.*, "Toward humanoid manipulation in human-centred environments," *Robot. Auton. Syst.*, vol. 56, pp. 54–65, 2008.
[25] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models database: An object model database for object recognition, localization and manipulation in service robotics," *Int. J. Robot. Res.*, vol. 31, no. 8, pp. 927–934, 2012.
[26] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. 2015 Int. Conf. Adv. Robot.* , Jul. 2015, pp. 510–517.