

On the Evolution of Fingertip Grasping Manifolds

K. Hang¹, J. A. Hausteин¹, M. Li², A. Billard², C. Smith¹ and D. Kragic¹

Abstract—Efficient and accurate planning of fingertip grasps is essential for dexterous in-hand manipulation. In this work, we present a system for fingertip grasp planning that incrementally learns a heuristic for hand reachability and multi-fingered inverse kinematics. The system consists of an online execution module and an offline optimization module. During execution the system plans and executes fingertip grasps using Canny’s grasp quality metric and a learned random forest based hand reachability heuristic. In the offline module, this heuristic is improved based on a grasping manifold that is incrementally learned from the experiences collected during execution. The system is evaluated both in simulation and on a Schunk-SDH dexterous hand mounted on a KUKA-KR5 arm. We show that, as the grasping manifold is adapted to the system’s experiences, the heuristic becomes more accurate, which results in an improved performance of the execution module. The improvement is not only observed for experienced objects, but also for previously unknown objects of similar sizes.

I. INTRODUCTION

Significant progress has been made in the area of robotic grasping [1]–[3]. For fingertip grasping, localizing contacts on the object that provide a stable grasp [4]–[6] and finding a hand configuration for realizing these contacts [7]–[9] have been addressed as separate problems. As a result, in these approaches it is not guaranteed that the planned contacts are kinematically feasible for a specific hand [1], [2].

To overcome this limitation, various grasp optimization frameworks have recently been proposed that integrate both stability and hand reachability analysis. In [10], the finger kinematics are modeled as an optimization constraint for efficiently finding contacts on objects that can be approximated by super-quadratics. Given a set of initial hand poses, [11] optimizes both hand configurations and contacts to generate a dense set of grasps offline. The framework presented in [12] plans contact triplets on incomplete 3D point clouds online, utilizing hand shape primitives as a heuristic to constrain the contacts optimization. In [13], a system first learns task related grasping parts of objects from human demonstrations. Next, the contact points on these parts are found by stochastic optimization, which is constrained by closed kinematic loops between the fingertips and the object part. In general, these systems require trade-offs between the desired computational efficiency and the precision of planning by either approximating the object shapes or the hand kinematics.

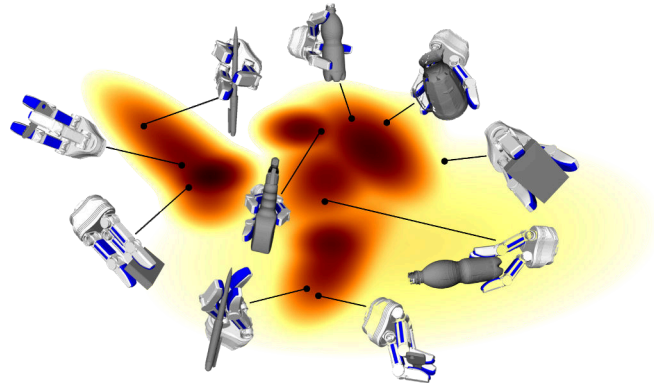


Fig. 1: A 2D-projection of the seven dimensional configuration space of the Schunk-SDH hand with example grasps. The colored areas show a grasping manifold that is learned from experiences. Video: <http://www.csc.kth.se/~kaiyuh/videos/graspManifold.mp4>.

In our recent work [14], we proposed a fast fingertip grasp planner that searches stable and reachable grasps in a hierarchical search space of fingertip contacts. To ensure kinematic feasibility for a dexterous redundant hand, an uniform sampling based affine-invariant hand reachability heuristic is used. However, as a common property of sampling based heuristics [15]–[18], the sampling resolution affects the efficiency and precision of this method. To achieve the same resolution, the number of required samples increases exponentially with the dimension of the configuration space. Further, in practice many of the samples cover invalid or task irrelevant configurations.

Instead of sampling randomly, learning from humans effectively excludes irrelevant hand configurations [19], [20]. However, this requires significant effort from human teachers and is limited by the teacher’s experience. Our key insight in this work is that we can limit the training set for a sampling based hand reachability heuristic to the manifold of for grasping relevant hand configurations. In doing so, we increase the heuristic’s accuracy in relevant regions of the configuration space, while reducing it in irrelevant regions, see Fig. 1. We propose that a robot should learn this grasping manifold based on its own task relevant experiences. The contributions of this work are:

- We present a learning framework that incrementally learns a subspace of the hand configuration space, a grasping manifold, that is relevant for fingertip grasping.
- We show that this allows us to incrementally train a heuristic for hand reachability and multi-fingered inverse kinematics, which improves as experiences are gained.
- We further show that we can extend this approach to learning a heuristic that integrates both grasp quality and hand reachability.

¹K. Hang, J. A. Hausteин, C. Smith and D. Kragic are with the Computer Vision and Active Perception Lab, CAS, CSC at KTH Royal Institute of Technology, Stockholm, Sweden. {kaiyuh, hausteин, ccs, dani}@kth.se.

²M. Li and A. Billard are with the Learning Algorithms and Systems Laboratory (LASA) at École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. {miaoli, aude.billard}@epfl.ch.

We define the terminology in Sec. II and introduce the proposed approach in Sec. III. In Sec. IV, we evaluate our approach both in simulation and on a real robot and end in Sec. V with a conclusion and a discussion of potential future work.

II. TERMINOLOGY

A. Fingertip Grasp and Hand Kinematics

In this work, we consider a dexterous multi-fingered robotic hand with d degrees of freedom and f fingertips. Let \mathcal{C} denote the d -dimensional configuration space of the hand and accordingly $\mathcal{C}_{free} \subset \mathcal{C}$ the set of self-collision-free hand configurations. For an object let $\mathcal{O} \subset \mathbb{R}^3$ denote the object's surface. A fingertip grasp $\gamma = (c_1, \dots, c_f)$ is a f -tuple of point contacts on the surface \mathcal{O} , where c_i is the contact of fingertip i with the object. Each contact $c_i = (p_i, n_i)$ consists of its position $p_i \in \mathcal{O}$ and normal $n_i \in \mathbb{S}^2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$ defined in the object's frame.

If the hand is in a configuration $\Theta \in \mathcal{C}_{free}$, the fingertip poses in the hand frame are $Z = (\zeta_1, \dots, \zeta_f) = F(\Theta)$, where F denotes the forward kinematics of the hand. Similar to a contact on \mathcal{O} , a fingertip pose is a tuple $\zeta_i = (p_i, n_i)$, where $p_i \in \mathbb{R}^3$ is the position and $n_i \in \mathbb{S}^2$ the normal of fingertip i in the hand frame. Let \mathcal{H} denote the pose of the hand frame relative to the object frame. A fingertip grasp γ is achieved by the hand configuration Θ , if $T_O(F(\Theta), \mathcal{H}) = \gamma$ holds. $T_O(Z, \mathcal{H})$ transforms a tuple of fingertip poses, Z , from the hand's frame to the object's frame.

B. Affine Invariant Grasp Encoding

We adopt an affine invariant grasp encoding as exemplified in Fig. 2 with $f = 3$ fingertips to easily determine whether a hand configuration can achieve a given grasp on an object. Given a fingertip grasp $\gamma = (c_1, c_2, c_3)$, we compute the grasp code as

$$C(\gamma) = (\|p_1 - p_2\|, \|p_2 - p_3\|, \|p_3 - p_1\|, \|n_1 - n_2\|, \|n_2 - n_3\|, \|n_3 - n_1\|). \quad (1)$$

Given a hand configuration with fingertip poses $Z = (\zeta_1, \zeta_2, \zeta_3)$, we compute $C(Z)$ in the same way as Eq. (1). A hand configuration Θ with $F(\Theta) = Z$ is feasible to achieve a grasp γ , if $C(\gamma) = C(Z)$. Note that this encoding allows us to check feasibility without explicitly computing \mathcal{H} .

C. Grasping Manifold

We wish to train a heuristic for hand reachability and multi-fingered inverse kinematics on the set of hand configurations that are relevant for fingertip grasping. Let \mathcal{G} denote the set of all stable fingertip grasps achievable by the robot hand on all objects. The manifold with corners [21] of hand configurations that achieve the fingertip grasps in \mathcal{G} is defined as

$$\mathcal{X} = \{\Theta \in \mathcal{C}_{free} | \exists \gamma \in \mathcal{G} : C(\gamma) = C(F(\Theta))\}. \quad (2)$$

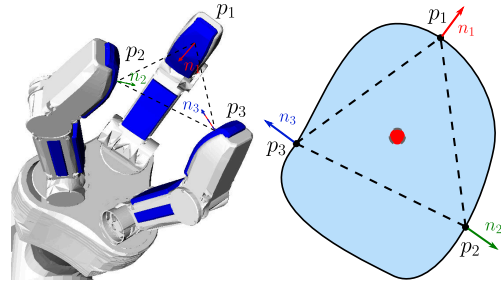


Fig. 2: The fingertip grasp encoding visualized for the Schunk-SDH hand and a 2D example object. The red dot shows the center of the contacts.

Because of the wide definition of \mathcal{G} , it is reasonable to focus on a robot specific subset

$$\bar{\mathcal{X}} = \{\Theta \in \mathcal{C}_{free} | \exists \gamma \in \mathcal{G}' : C(\gamma) = C(F(\Theta))\} \subset \mathcal{X}, \quad (3)$$

where $\mathcal{G}' \subset \mathcal{G}$ is the set of fingertip grasps on the objects the robot at hand will actually encounter during its lifespan.

Computing $\bar{\mathcal{X}}$ as well as \mathcal{G}' explicitly, however, is infeasible. Further, the set of objects a robot may encounter during its lifespan may not be known beforehand. Hence, the system we present provides an incrementally improved estimate of $\bar{\mathcal{X}}$ based on its own experiences. Throughout this work we denote such an estimate of $\bar{\mathcal{X}}$ as *grasping manifold*.

III. METHODOLOGY

An overview of our system is given in Fig. 3. It consists of an *execution* module, an offline optimization module denoted as *dreaming* module, a *memory* and a heuristic \mathcal{R} . The execution module plans and executes grasps and, if successful, feeds its experiences to the memory. The dreaming module takes these experiences from the memory and evolves a grasping manifold, which is used as training set for the heuristic \mathcal{R} . As the system acquires a more specialized grasping manifold, i.e. a better estimate of $\bar{\mathcal{X}}$, the heuristic \mathcal{R} becomes more accurate and in turn facilitates the execution module in the future.

In the following of this section, we first explain the execution module in relation to our previous work and discuss how our heuristic affects its performance. Thereafter, we introduce the evolution of the grasping manifold and describe how the system utilizes this to learn \mathcal{R} . In the end, we propose an extension to our system that is able to generate stable grasps purely from learned experiences.

A. Execution Module

We adopt the grasp planner presented in our previous work [14], which computes a grasp γ by maximizing the objective function

$$\Gamma(\gamma) = Q(\gamma) + \alpha R(\gamma) \quad (4)$$

in the Hierarchical Fingertip Space (HFTS). Here, $Q(\gamma) \in \mathbb{R}$ denotes the point contact based grasp quality measure Q_1 from [4], $R(\gamma) \in \mathbb{R}$ the reachability residual that measures the kinematic feasibility of realizing the fingertip grasp γ with the robot hand and $\alpha \in \mathbb{R}$ a scaling factor.

In [14], $R(\gamma)$ is computed by querying a kd-tree based data structure for a stored hand configuration Θ_γ that

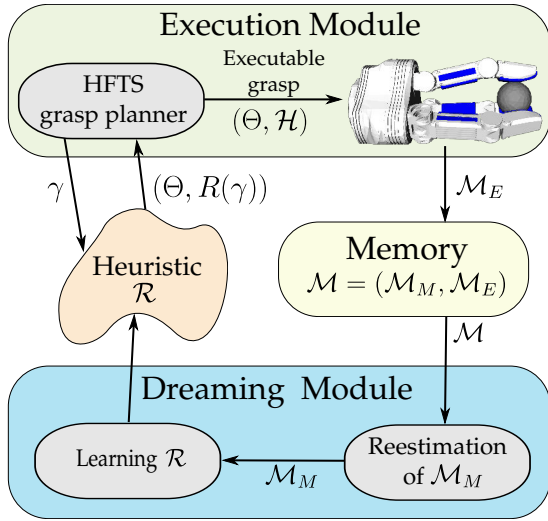


Fig. 3: System pipeline: The system consists of an execution module, an offline module, denoted dreaming module, a memory \mathcal{M} and a heuristic \mathcal{R} .

minimizes the distance $\|C(F(\Theta_\gamma)) - C(\gamma)\|$. In general, due to its limited sample density, for an arbitrary input grasp γ there is no hand configuration Θ_γ stored in the data structure that achieves an exact reachability residual of zero, i.e. $\|C(F(\Theta_\gamma)) - C(\gamma)\| \neq 0$. Therefore, after determining a grasp γ^* that maximizes Γ , the planner is required to perform a post-optimization step to adjust the hand configuration Θ_{γ^*} to minimize the residual. Finally, the hand pose \mathcal{H} is computed such that the fingertip positioning error $\epsilon = \|\gamma^* - T_O(F(\Theta_{\gamma^*}), \mathcal{H})\|$ is minimized. Note that the computation of \mathcal{H} and Θ_{γ^*} may fail in case γ^* is not reachable. In this case the system needs to re-plan.

In this work in contrast to [14], we replace the kd-tree based heuristic by our heuristic \mathcal{R} . Similar to our previous heuristic, \mathcal{R} is defined as

$$\mathcal{R} : \gamma \mapsto (\Theta_\gamma, R(\gamma)) \quad (5)$$

and provides a reachability estimate $R(\gamma) \in \mathbb{R}$ of how well a fingertip grasp γ is kinematically reachable, as well as an inverse kinematics solution Θ_γ such that $C(F(\Theta_\gamma)) \approx C(\gamma)$.

Every time the system successfully computed and executed a grasp on some object, the post-optimized hand configuration for this grasp is stored in the memory \mathcal{M}_E . In the offline dreaming phase, the system recomputes its grasping manifold based on the collected experiences. As the heuristic is retrained on the updated grasping manifold, it provides a better estimate of hand reachability and hand inverse kinematics.

B. Experience Oriented Manifold Evolution

The dreaming phase is summarized in Algorithm 1. We represent the grasping manifold by a discrete set of N_i hand configuration samples $\mathcal{M}_M \subset \mathcal{C}_{free}$, which is stored in the system's memory \mathcal{M} . Before any experiences have been gathered, there is no knowledge about $\bar{\mathcal{X}}$. Hence, it is assumed that any self-collision-free hand configuration may lie within it and we initialize our initial estimation from a uniform distribution, see Algorithm 1 line 2. At this point,

Algorithm 1: The *dreaming* phase: Evolution of grasping manifold

Input: Memory $\mathcal{M} = (\mathcal{M}_M, \mathcal{M}_E)$
Constants: Covariance σ , Number of manifold samples N_i , Sample parameters β and ρ
Output: Updated memory $\mathcal{M} = (\mathcal{M}_M, \mathcal{M}_E)$, Heuristic \mathcal{R}

```

1 if  $\mathcal{M}_M = \emptyset$  then
2   // Bootstrap
3    $\mathcal{M}_M \leftarrow \text{SAMPLEUNIFORM}(\mathcal{C}_{free}, N_i)$ 
4 else
5   // Evolution
6    $\Omega \leftarrow \text{FITGMM}(\mathcal{M}_M)$ 
7    $S_n \leftarrow \emptyset$ 
8   for  $\Theta_E \in \mathcal{M}_E$  do
9      $N_r \leftarrow \frac{\beta|\mathcal{M}_M|}{|\mathcal{M}_E|} (\frac{1}{\rho m} - \int_{\mathcal{C}} g(\Theta|\Theta_E, \sigma I) p(\Theta|\Omega) d\Theta)$ 
10     $N_r \leftarrow \max(N_r, 0)$ 
11     $S_n \leftarrow S_n \cup \text{SAMPLEGAUSSIAN}(\Theta_E, \sigma I, N_r)$ 
12   $\mathcal{M}_M \leftarrow \text{SAMPLEUNIFORM}(S_n \cup \mathcal{M}_M, N_i)$ 
13   $\mathcal{M}_E \leftarrow \emptyset$ 
14   $\mathcal{R} = \text{COMPUTER}(\mathcal{M}_M)$ 

```

\mathcal{R} can be learned as detailed in Sec. III-C and the system is similar to our previous work [14] and ready for execution.

Once the robot gathered some experiences and enters the *dreaming* phase anew, we update \mathcal{M}_M to more accurately approximate $\bar{\mathcal{X}}$. This update is performed by replacing samples in \mathcal{M}_M with self-collision-free hand configurations scattered around the experiences stored in \mathcal{M}_E . For each experience Θ_E , new samples are sampled from a Gaussian distribution $\mathcal{N}(\Theta_E, \sigma I)$ centered at Θ_E with a user specified diagonal covariance matrix $\sigma I \in \mathbb{R}^{d \times d}$, see Algorithm 1 line 9. The covariance controls the adaptation of the manifold. A large value leads to a diffuse, whereas a smaller value to a more concentrated adaptation. The motivation for sampling a neighborhood of Θ_E lies in the fact that the grasp quality measure Q is Lipschitz continuous [22]. Hence, since an experience Θ_E is a hand configuration that achieves a stable grasp on some object with fingertip poses Z , it is reasonable to assume that fingertip poses similar to Z can also achieve good quality grasps on similarly shaped objects.

We need, however, to balance between evolving the grasping manifold towards new experiences and preserving the manifold learned so far. For this, we compute a probability distribution $p(\Theta|\Omega)$, denoting the probability of a configuration $\Theta \in \mathcal{C}$ lying in $\bar{\mathcal{X}}$ given our current grasping manifold. We model $p(\Theta|\Omega)$ as Gaussian Mixture Model (GMM) and use the expectation maximization (EM) algorithm to compute the GMM parameters Ω from our sample set \mathcal{M}_M , see Algorithm 1 line 4. As shown in Algorithm 2 we select Ω such that the Bayesian information criteria (BIC) is minimized.

In order to adapt the grasping manifold, the system determines for each $\Theta_E \in \mathcal{M}_E$ its novelty utilizing $p(\Theta_E|\Omega)$. Experiences that are likely given our current grasping manifold should result in little adaptation, whereas experiences that are unlikely are considered novel and should result in significant adaptation. The degree of adaption of the grasping manifold to an experience Θ_E is governed by the number of

Algorithm 2: FITGMM: Determine parameters Ω for a set of sampled configurations

Input: S set of samples

Output: Best fitting Ω

```

1  $BIC = \infty$ 
2 for  $m = 1 \dots \hat{m}$  do
3    $\Omega' = \text{EMALGORITHM}(S, m)$ 
4   if  $\text{BIC}(\Omega') < BIC$  then
5      $\Omega = \Omega'$ 
6      $BIC = \text{BIC}(\Omega)$ 
7 return  $\Omega$ 

```

samples $N_r \in \mathbb{N}^+$ we sample in its neighborhood. Given the GMM parameters Ω with m Gaussians, N_r is computed as follows:

$$N_r = \max\left(\frac{|\mathcal{M}_M|}{|\mathcal{M}_E|} \left(\frac{1}{\rho m} - \int_{\mathcal{C}} g(\Theta|\Theta_E, \sigma I) p(\Theta|\Omega) d\Theta\right), 0\right) \quad (6)$$

where $g(\Theta|\Theta_E, \sigma I)$ is the probability density of the Gaussian $\mathcal{N}(\Theta_E, \sigma I)$, and $p(\Theta|\Omega)$ is the probability density of the GMM parametrized by Ω . $\beta \in \mathbb{R}$ controls the evolution speed and $\rho \geq \beta$ is a relaxation parameter. N_r is large, if the probability of samples in the neighborhood of Θ_E belonging to the current grasping manifold is small and vice versa. The term $\frac{1}{\rho m}$ in Eq. (6) denotes the probability mass we desire this neighborhood to have.

Although the dreaming phase is performed offline, we choose to limit its computational effort by performing batch updates rather than updating \mathcal{M}_M for each experience $\Theta_E \in \mathcal{M}_E$ individually. In the rare case that all experiences lie in the same small area of \mathcal{C}_{free} , this could lead to a rapid adaptation of our manifold to a single mode. To prevent this, N_r decreases as $|\mathcal{M}_E|$ increases and vice versa.

Once all samples S_n around all new experiences are sampled, we recompute the grasping manifold by uniformly re-sampling N_i samples from $S_n \cup \mathcal{M}_M$, see Algorithm 1 line 10. Note that while we desire \mathcal{M}_M to resemble $\bar{\mathcal{X}}$, the proposed method does not guarantee that the estimate \mathcal{M}_M lies strictly within $\bar{\mathcal{X}}$. However, as the evolution progresses, the probability of including outliers in \mathcal{M}_M decreases.

C. Reachability Heuristic by Regression Forest

Our reachability heuristic is a function as defined in Eq. (5), which maps a fingertip grasp γ to a hand configuration Θ_γ as well as the probability $R(\gamma)$ that $C(\gamma) = C(F(\Theta_\gamma))$. For modeling this high dimensional non-linear mapping, we adopt the regression forest from [23] consisting of T regression trees. In order to maximize \mathcal{R} 's accuracy for relevant hand configurations, the grasping manifold \mathcal{M}_M serves as basis for the training set. Since fingertip grasps are defined in an object frame and therefore object specific, we learn a mapping from the encodings $C(\gamma)$ to hand configurations, making \mathcal{R} object frame independent. The labeled training set for the regression forest is

$$\mathcal{T} = \{(\Theta, C(F(\Theta))) | \Theta \in \mathcal{M}_M\}. \quad (7)$$

Each leaf of each tree in the forest provides a prediction model that is learned by probabilistic linear regression with a Gaussian distribution [24] from some subset of \mathcal{T} . In order to achieve a fully probabilistic output, the splits of the training set within each regression tree are performed such that the information gain reported in [23] is maximized.

Given an input grasp γ , the t -th tree in the forest, $t = 0, \dots, T - 1$, provides the posterior $p_t(\Theta|C(\gamma))$. The posterior of the whole forest is the average of the posteriors of the individual trees

$$p(\Theta|C(\gamma)) = \frac{1}{T} \sum_{t=1}^T p_t(\Theta|C(\gamma)). \quad (8)$$

With the posterior at hand we can compute the heuristic value $\mathcal{R}(\gamma) = (\Theta_\gamma, R(\gamma))$ as

$$\Theta_\gamma = \mathbb{E}[\Theta|C(\gamma)] = \int_{\mathcal{C}} p(\Theta|C(\gamma)) \Theta d\Theta, \quad (9)$$

$$R(\gamma) = p(\Theta_\gamma|C(\gamma)). \quad (10)$$

Note that the evaluation of $\mathcal{R}(\gamma)$ is computationally inexpensive and thus suitable for grasp optimization.

D. Pure Manifold Based Grasp Planning

The planner as presented in Sec. III-A computes a grasp by maximizing both the reachability of a fingertip grasp as well as its quality, Eq. (4). As reported in [4], [5], the scaling factor $\lambda \in \mathbb{R}^+$ between the force term and the torque term in the computation of the grasp quality $Q(\gamma)$ is somehow arbitrarily chosen. With a large value, the grasp will be more optimized to counteract torsional disturbances, whereas a small value makes the grasp to be more optimized towards translational disturbances. Additionally, as we do not have an accurate prior of friction coefficients, the planned grasps can be vulnerable to friction changes [25].

Recall that the evolution of the grasping manifold is based on successful stable grasps. Therefore, during the evolution, the system learns not only the relevant grasp configurations, it also implicitly suppresses grasps that are not robust, i.e. grasps that are not always stable during execution. As an extension to our system, we therefore propose a modification of our heuristic \mathcal{R} that not only provides a reachability estimate, but also a quality estimate. Formally, we define the modified heuristic as

$$\mathcal{R}^* : (r, C(\gamma)) \mapsto (\Theta_\gamma, R^*(\gamma)). \quad (11)$$

Additional to the contacts encoding $C(\gamma)$, \mathcal{R}^* takes the distance $r \in \mathbb{R}^+$ between the center of mass of the object and the center of the contact positions as argument.

In the execution module the system initially uses \mathcal{R} as described before. When a grasp γ is successfully executed, we save both r and Θ_γ in an additional memory \mathcal{M}_E^* . When the grasping manifold evolution becomes stable, i.e. S_n is small, the grasping manifold provides a good estimate of $\bar{\mathcal{X}}$. In this case, we train \mathcal{R}^* using the same method as in Sec. III-C with r as additional variable in the input space. The training set in this case is

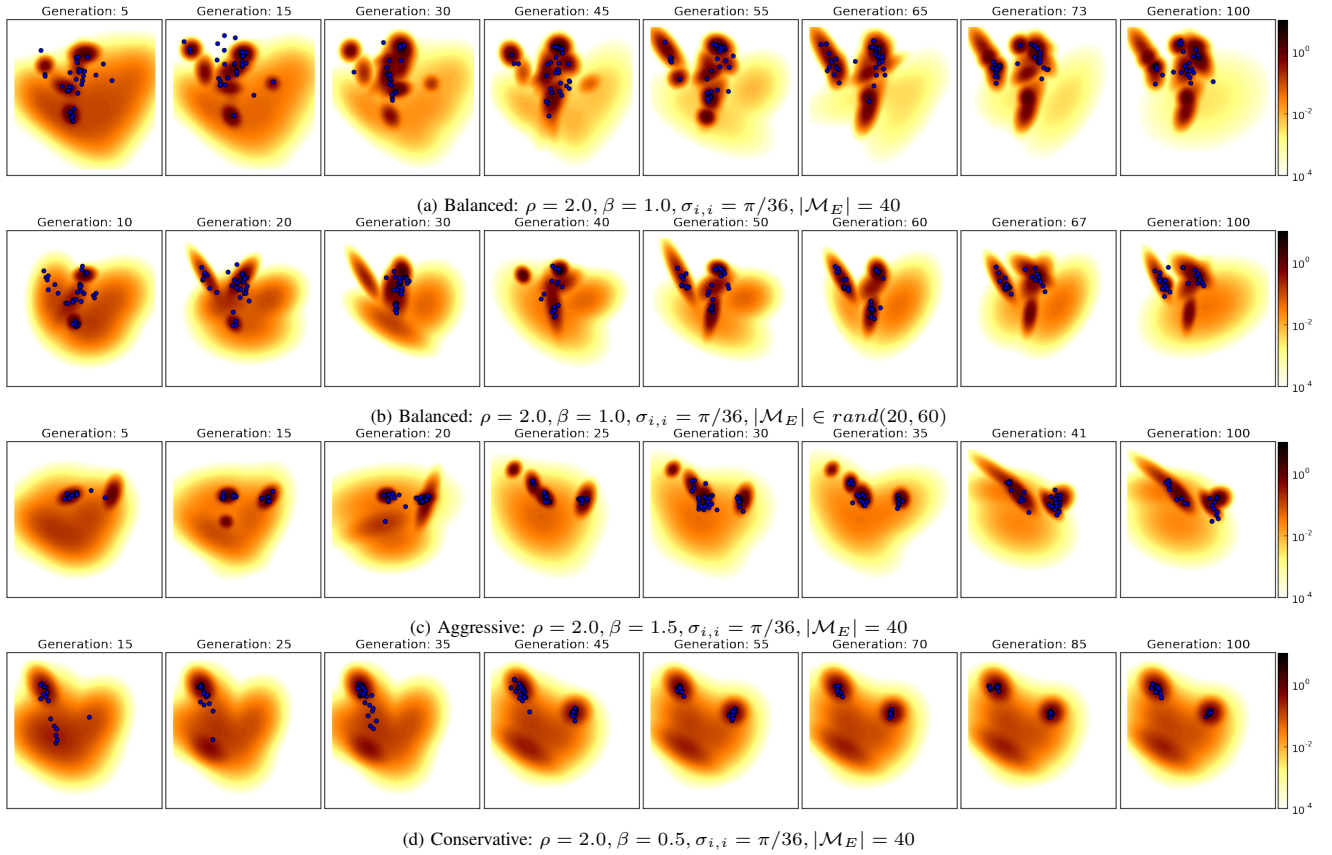


Fig. 4: Evolution examples with different evolution speeds β or $|\mathcal{M}_E|$. The two dimensional visualization is a projection of \mathcal{M}_M using the first two principal components. In order to compare between different evolutions, all projections are conducted under a single PCA on the 100-th generation \mathcal{M}_M shown in (a), which captures 78% of the variance. The blue marks are the grasp experiences \mathcal{M}_E used to generate the current generation. The evolutions took on average 5–10min per generation.

$$\mathcal{T}^* = \{(\Theta, (C(F(\Theta)), r)) | (\Theta, r) \in \mathcal{M}_E^*\}. \quad (12)$$

Henceforth, we use \mathcal{R}^* as a generative model: instead of explicitly computing the grasp quality $Q(\gamma)$, the objective function Γ for the HFTS grasp planner becomes the posterior $R^*(\gamma)$ of the \mathcal{R}^* prediction. Since \mathcal{R}^* is not generative for new types of objects, the system falls back to \mathcal{R} and re-enables its evolution to adapt to the new types of objects, when these are encountered.

IV. EXPERIMENTS

We implemented the proposed system in Python and evaluate it for the Schunk-SDH hand with $d = 7$ DoFs and $f = 3$ fingertips in the OpenRAVE simulation environment [26] on a machine with an Intel Core i7-3770 CPU @ 3.40GHz \times 8 and 32GB RAM. In the execution module, when planning grasps using the HFTS grasp planner [14], the maximum iteration per hierarchy level is set to 40. In the dreaming module, the grasping manifold is represented by $N_i = |\mathcal{M}_M| = 10^5$ samples and the GMM model Ω is estimated using full covariance matrices. For the construction of the random forest, the forest size is set to $T = 10$, the maximum tree depth is set to 10 and the minimum samples for splitting a decision node is 20.

We first evaluate the evolution of the grasping manifold using experiences collected in simulation. Thereafter, we

present quantitative results to evaluate the extension described in Sec. III-D. Finally, we show real world experiments with a Schunk-SDH hand mounted on a KUKA KR5 sixx 850 arm.

A. Evolution of Grasping Manifold

For the evaluation of the grasping manifold evolution, we investigate whether we can observe the following properties:

- **P.1** As the grasping manifold evolves, the number $|\mathcal{S}_n|$ of new samples drawn from the neighborhoods of the gathered experiences decreases. A larger evolution speed β results in a faster decrease in $|\mathcal{S}_n|$, but less generality across different grasp types and vice versa.

- **P.2** As the grasping manifold evolves, \mathcal{R} provides better predictions of hand configurations for grasps on the training objects as well as on unexperienced objects of similar sizes.
- **P.3** A grasping manifold that is learned from experiences on some objects continues evolving towards new experiences once it is exposed to objects requiring significantly different grasps. By doing so, it generalizes over different objects and \mathcal{R} 's performance for previously experienced objects does not deteriorate.

1) *Investigation of P.1:* Fig. 4 shows the evolution of four different grasping manifolds with different choices for the parameters β and $|\mathcal{M}_E|$. In all cases the system gathers experiences by generating grasps on the two objects *rivella*

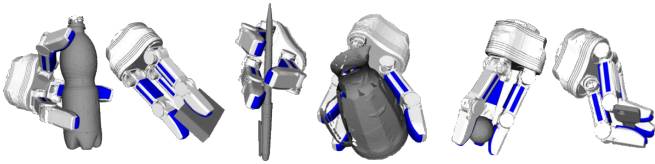


Fig. 5: Examples grasps on the objects used in evaluation, from left to right: rivella, box, pen, jug, ball, key.

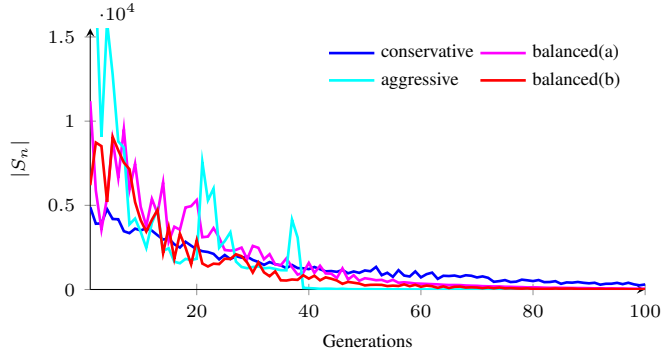


Fig. 6: Number of new samples $|S_n|$ of the evolution examples shown in Fig. 4.

and *box* shown in Fig. 5. We initialize the grasping manifolds with $N_i = 10^5$ uniformly sampled collision-free hand configurations. Except for the evolution shown in Fig. 4b, our system enters the dreaming phase every time it has experienced $|\mathcal{M}_E| = 40$ stable grasps (evaluated in simulation using [4]), i.e. 20 per object. In case of the evolution shown in Fig. 4b, it enters the dreaming phase every time a random number $|\mathcal{M}_E| = n_r + n_b$ of experiences has been made, where $n_r, n_b \in [10, 30]$ denote the number of experiences on *rivella* and *box* respectively. The parameter β is chosen such that in Fig. 4a and Fig. 4b we expect a balanced evolution, whereas in Fig. 4c we expect an aggressive evolution towards new experiences and in Fig. 4d a conservative evolution that slowly adapts to new experiences.

As shown in Fig. 6, we observe that the number of new samples in the evolution process is decreasing for all four evolutions. The first three evolutions shown in Fig. 4 reach a smaller number of new samples faster than the conservative evolution in Fig. 4d. We can observe visually in Fig. 4 that after generations 73, 67, 41 and 85 the grasping manifolds do not change significantly anymore.

It is worth noticing that the more aggressive the evolution is, the faster it achieves a stable manifold. However, as we can see from the evolution in Fig. 4c, an aggressive evolution results in a grasping manifold that is concentrated on few areas, which is where early experiences originate from. The conservative evolution in Fig. 4d takes the longest to achieve a stable grasping manifold, however, it is more spread. The balanced evolutions achieve a greater variety of modes than the aggressive evolution, while reaching a stable grasping manifold in less generations than the conservative one. The balanced manifold from Fig. 4a is also shown in Fig. 1, which shows the diversity of grasp configurations it contains. In contrast, the aggressive evolution suppresses the variety of grasps the system can generate. In summary, the above

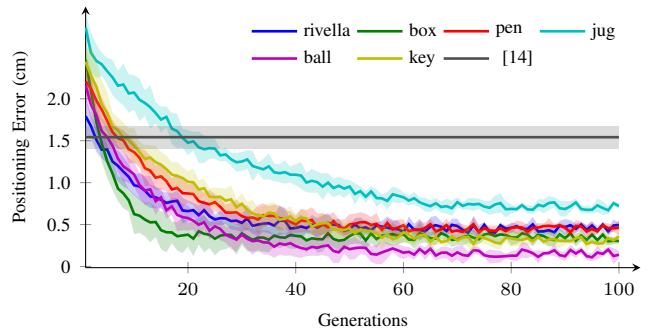


Fig. 7: Positioning errors ϵ for the evolution shown in Fig. 4(a). The gray line shows the errors of the heuristic reported in [14]. The reported results are collected from 100 simulated grasps on each object per generation.

results support our assumption about property **P.1**.

2) *Investigation of P.2:* Fig. 7 shows the development of the positioning error $\epsilon = \|\gamma - T_O(F(\Theta), \mathcal{H})\|$ between fingertips and the desired contacts on an object surface for different objects as the evolution of the balanced manifold from Fig. 4a progresses. We can see that before the evolution starts, our heuristic performs worse than the reachability heuristic proposed in [14], since the linear regression performed by the random forest performs poorly for sparse data points. As the evolution proceeds, however, the positioning error decreases for all six objects. Note that the evolution is performed with the objects *rivella* and *box*, for the other objects we only evaluate \mathcal{R} without feeding the experiences to the dreaming module. Nevertheless, the error also reduces for these objects, indicating that the system is capable of generalizing. This result supports our assumption about property **P.2:** as the evolution of the grasping manifold progresses, \mathcal{R} 's performance increases both for previously seen and unseen objects of similar sizes.

3) *Investigation of P.3:* For evaluating property **P.3**, we run the evolution in Fig. 4a again with a different setup: as shown in Fig. 8, once the evolution for *rivella* and *box* is stable at generation 73, we ask the system to additionally plan grasps for a new object *rivella_{big}*, which is the *rivella* bottle scaled up by a factor of 4.0. An object of this size requires grasps for which the hand needs to be opened far wider than for any of the others. We can see that since the system has not experienced an object of this scale before, the positioning error is initially large for *rivella_{big}*. However, as first experiences are made, the system starts to evolve anew, adapts its grasping manifold and eventually becomes stable again at generation 99. Thereafter, by asking it to plan grasps for a large disk, which requires the coupled joint of the Schunk-SDH hand to open even further, the system evolves further until generation 126. It is worthwhile to note that when the new objects are introduced to the system, the positioning errors for the previously experienced objects do not increase. In summary, these results support our assumption about property **P.3**.

B. Evaluation of \mathcal{R}^*

As described in Sec. III-D, once the grasping manifold evolution becomes stable, we can learn a new heuristic \mathcal{R}^*

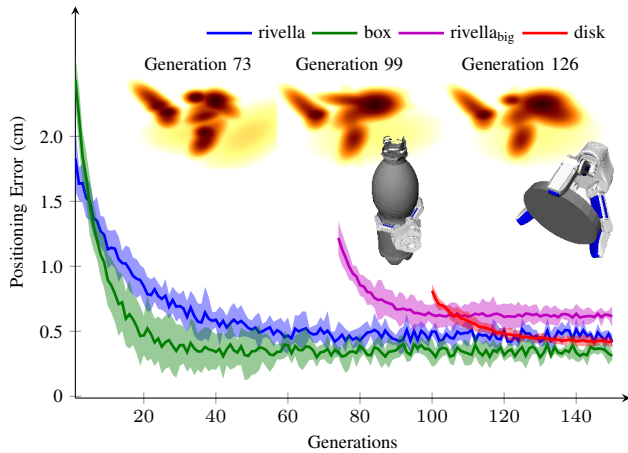


Fig. 8: Positioning errors ϵ , the evolution is controlled by the parameters shown in Fig. 4(a). Once the evolution becomes stable for *rivella* and *box*, the new objects *rivella_{big}* and *disk* are introduced into the system and trigger the continuation of the evolution. The reported results are collected from 100 simulated grasps on each object per generation.

from all gathered experiences. Next, we investigate whether we can observe the following property:

- **P.4** Once the grasping manifold becomes stable, our grasp planner with learned heuristic \mathcal{R}^* is able to plan stable grasps with a performance equal to the planner with heuristic \mathcal{R} , while achieving faster runtimes.

Object	Err(cm)	Q	Time(s)
rivella	0.47 ± 0.07	1.13 ± 0.24	0.95 ± 0.06
	0.42 ± 0.04	1.13 ± 0.09	0.37 ± 0.06
box	0.39 ± 0.04	1.24 ± 0.37	0.79 ± 0.02
	0.37 ± 0.06	1.22 ± 0.13	0.36 ± 0.07
pen	0.55 ± 0.02	0.20 ± 0.04	0.99 ± 0.07
	0.57 ± 0.03	0.22 ± 0.02	0.29 ± 0.01
jug	0.85 ± 0.13	1.27 ± 0.22	1.21 ± 0.27
	0.83 ± 0.07	1.26 ± 0.14	0.33 ± 0.03
ball	0.09 ± 0.01	0.42 ± 0.02	0.94 ± 0.04
	0.10 ± 0.01	0.42 ± 0.01	0.22 ± 0.01
key	0.65 ± 0.06	0.22 ± 0.02	1.14 ± 0.23
	0.52 ± 0.04	0.23 ± 0.04	0.32 ± 0.04

Fig. 9: Statistics for the comparison between \mathcal{R} and \mathcal{R}^* (shaded). The results reported are collected by generating 100 grasps in simulation for each object for both heuristics respectively. Err: positioning errors, Q: grasp quality measured by [4], Time: runtime for planning a grasp. The evaluations were implemented in Python and run on a machine with Ubuntu 12.04 running on an Intel Core i7-3770 @ 3.40GHz \times 8. with 32GB RAM

When the grasping manifold evolution shown in Fig. 4a is stable at generation 73, the system has experienced $73 \times 2 \times 20 = 2920$ grasps. Hereafter, the system uses the 2920 experiences to learn the new heuristic \mathcal{R}^* . From the table shown in Fig. 9 we can see that the mean positioning errors and the grasp qualities for both heuristics are similar. This is expected since the stability manifold learned from the 2920 experiences tends to lead the planner to plan grasps similar to its experiences. Note that as the computation of the grasp quality metric is omitted, the runtime for the grasp planner using \mathcal{R}^* is in general decreased by 2/3 to 3/4. This result supports our assumption about property **P.4**.

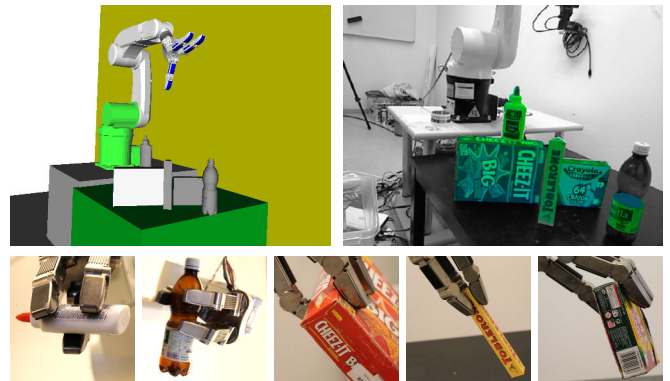


Fig. 10: *Top*: As experimental setup we utilize a Schunk-SDH hand mounted on a KUKA-KR5 industrial arm. For localizing objects in the robot’s workspace, we use SimTrack [27] in combination with a Microsoft KinectTM. *Bottom*: Stable fingertip grasps executed on the five different objects used in our experiments.

C. Evaluation on a Real Robot

Finally, we wish to ensure that our system produces grasps that can be executed successfully on a real robot. In particular, we assume the following property:

- **P.5** Grasps generated with the learned heuristic \mathcal{R}^* are stable in the real world.

For this, we execute grasps on the system illustrated in Fig. 10. The grasp planner uses \mathcal{R}^* which was trained on the shown five objects in simulation. For each experiment, we place one of the objects at an arbitrarily chosen reachable pose on a table in front of the robot. The robot is tasked with picking up the object and placing it in a bin located next to the table. We execute this sequence for each object six times and record the number of successful placements. As our grasp planner does not consider collisions with the environment nor the kinematics of the robot arm, we repeatedly plan grasps anew until a reachable grasp has been found. To execute grasps, we utilize a position controller for the finger joints. To exert some force on the grasped object, we additionally close the fingers approximately 0.1cm along the contact normals.

Object	#Trials	#Success	Notes
rivella	7.3 ± 9.1	5/6	1 unstable
glue	9.5 ± 6.0	6/6	
crayola	3.0 ± 3.95	6/6	
toblerone	12.2 ± 10.69	4/6	tip over & unstable
cheezit	11.0 ± 9.7	6/6	

Fig. 11: Statistics of running \mathcal{R}^* on a real robot system, where six grasps were tested on each object. #Trials: number of grasps planned until a reachable grasp was found. #Success: number of successful pick-and-place executions. Notes: reason of failures.

The results are summarized in Fig. 11. In total, we only observed three failures. In two cases the executed grasps were unstable and the objects slipped, in the third case the object tipped over before all contacts were reached. Overall, the results support our assumption that the system has property **P.5**. However, in most cases the planner had to be executed several times to find a grasp that was reachable by the robot. This highlights a weakness of the current state of the proposed system. For fast fingertip grasp planning, a

system needs to take an object's environment as well as the robot arm's kinematics into account.

V. DISCUSSION & CONCLUSION

In this work, we presented a system that integrates a random forest based heuristic for hand reachability and multi-finger inverse kinematics with the fingertip grasp planner presented in [14]. The training set of the heuristic, the grasping manifold, is incrementally adapted by the system based on the experiences it gathered online. This allows to focus the heuristic's limited accuracy on the regions of the robot hand's configuration space that are task relevant. We showed that our system creates grasping manifolds that facilitate the performance of the grasp planner, while being able to generalize and continuously adapt to new types of experiences.

We further extended the proposed system to learn a heuristic from a grasping manifold that integrates both grasp quality and hand reachability, rendering the use of an analytic grasp quality metric unnecessary. As a result, we show that the planner's runtime is reduced, while it is still capable of generating stable grasps.

The system was evaluated both in simulation as well as on a real robot, whereas training was only performed in simulation. Ideally our system would learn its grasping manifold from experienced grasps that were successful in the real world. However, our current implementation of the system does not take the environment nor the kinematics of the robot's arm into account. As a result, the planner often produces grasps that are not kinematically reachable by the arm or in collision with the environment. Therefore, since the training requires a large number of executions, our evaluation on the real robot was limited to testing a system that was trained in simulation. In future work, we wish to explore the system's performance when its experiences are gathered in the real world. As the grasping manifold is learned from successfully executed grasps, it would implicitly emphasize grasps that have been stable and suppress grasps that have falsely been predicted to be stable. This may occur due to errors in the used model, such as the simplified modeling of contacts and unknown friction coefficients. A grasping manifold trained from real world examples could overcome these limitations of analytic grasp quality metrics. Moreover, we would like to look into the scalability of the proposed method with respect to the DoFs of the hand and the range of different object shapes it is able to cover. Here, it is interesting to investigate the possibility of adaptively evolving the number of samples used to represent the grasping manifold.

VI. ACKNOWLEDGMENTS

This work was supported by the European Union framework program FP8-645403 RobDREAM and FP7-288533 ROBOHOW.COGE.

REFERENCES

- [1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis – a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.
- [2] A. Sahbani, S. El-Khoury, and P. Bidaud, "An Overview of 3D Object Grasp Synthesis Algorithms," *Robot. Auton. Syst.*, 2011.
- [3] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000.
- [4] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992.
- [5] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2003.
- [6] M. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-d objects," *IEEE Transactions on Robotics*, 2009.
- [7] Y. Liu, H. Wang, and W. Zhou, "Kinematic feasibility analysis of 3-d multifingered grasp," in *Control and Decision Conference*, 2009.
- [8] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2002.
- [9] C. Rosales, J. Porta, R. Suarez, and L. Ros, "Finding all valid hand configurations for a given precision grasp," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008.
- [10] S. El Khoury, M. Li, and A. Billard, "Bridging the gap: One shot grasp synthesis approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct 2012.
- [11] J.-P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand," *Robot. Auton. Syst.*, 2012.
- [12] I. Gori, U. Pattacini, V. Tikhonoff, and G. Metta, "Three-finger precision grasp on incomplete 3d point clouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014.
- [13] S. El-Khoury and A. Sahbani, "A new strategy combining empirical and analytical approaches for grasping unknown 3d objects," *Robot. Auton. Syst.*, 2010.
- [14] K. Hang, J. A. Stork, and D. Kragic, "Hierarchical fingertip space for multi-fingered precision grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2014.
- [15] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2009.
- [16] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, "Bimanual grasp planning," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2011.
- [17] K. Hertkorn, M. Roa, and B. Ch, "Planning in-hand object manipulation with multifingered hands considering task constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013.
- [18] F. Zacharias, C. Borst, and G. Hirzinger, "Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009.
- [19] C. Fernandez, O. Reinoso, A. Vicente, and R. Aracil, "Kinematic redundancy in robot grasp synthesis. an efficient tree-based representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005.
- [20] E. L. Sauser, B. D. Argall, G. Metta, and A. G. Billard, "Iterative learning of grasp adaptation through human corrections," *Robot. Auton. Syst.*, 2012.
- [21] D. Joyce, "On manifolds with corners," in *'Advances in Geometric Analysis', Advanced Lectures in Mathematics 21, International Press, Boston*, 2012.
- [22] F. T. Pokorny and D. Kragic, "Classical grasp quality evaluation: New theory and algorithms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013.
- [23] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, 2012.
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
- [25] K. Hang, F. T. Pokorny, and D. Kragic, "Friction coefficients and grasp synthesis," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2013.
- [26] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, 2010.
- [27] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Hamburg, Germany, 2015.