Learning and Inference of Dexterous Grasps for Novel Objects with Underactuated Hands

Marek Kopicki[1] and Carlos J. Rosales[2] and Hamal Marino[2] and Marco Gabiccini[2] and Jeremy L. Wyatt[1]

Abstract— **Recent advances have been made in learning of grasps for fully actuated hands. A typical approach learns the target locations of finger links on the object. When a new object must be grasped, new finger locations are generated, and a collision free reach-to-grasp trajectory is planned. Such a division of labour fails to transfer directly to underactuated hands, which improve grasp reliability via contacts with the object during grasping. In this paper we present a method for learning transferrable grasps for underactuated hands. Our approach learns not only the desired final grasp, and also good grasping trajectories, from a rigid body simulation. This enables us to learn how to approach the object and close the underactuated hand from a variety of poses. Our method does not rely on explicit representation of the contact sequence. The core learning method uses product of experts. This allows grasp transfer to novel objects and works despite partial shape reconstruction of less than 25% of the surface. From nine training grasps on three objects the method transferred grasps to previously unseen, novel objects, that differ significantly from the training objects, with an 80% success rate. We move beyond previous work by: i) showing the ability to learn transferrable grasps for underactuated hands; ii) extending our learning method to work with multiple training examples for each grasp type; iii) extending our method to work with multiple reach to grasp trajectories.**

I. INTRODUCTION

Transferring dexterous grasps to novel objects is a challenging problem. One approach is to machine learn solutions with techniques able to perform powerful generalisation. Another is to use an underactuated hand to cope with shape variation. In this paper we combine the benefits of both approaches by learning grasps for underactuated hands. Underactuated hands exploit the contacts that occur during grasping to achieve a wide variety of final grasp configurations. The final grasp configuration depends not only on the final hand pose, but also on the object shape, and on the reach to grasp trajectory. An interesting challenge is to use machine learning to exploit these interactions. The key technical challenge in applying machine learning to grasping with underactuated hands is to learn the right trajectory for a particular object shape so as to achieve a good grasp of a particular type.

One approach would be to learn the typical contact interactions that occur during a grasp, and to generate new grasps that reproduce these. The contact interactions are, however, rather complex and variable, even given small variations in object shape and friction. Therefore we tackle the problem by implicitly encoding the contact interactions in terms of the approach trajectory. Our method learns both the desired final contacts, the final hand shape, and possible sequences of hand pose during the reach to grasp trajectory. We build on our previous work on one-shot learning of grasps that transfer to novel objects, employing a product of experts. The novel technical contribution is that in this paper we show how to learn not only final grasp, but also the approach trajectory and control strategy for closing the dexterous hand. In particular, we learn a bundle of multiple trajectories that will all likely lead to a similar final stable grasp. We enable this by learning from examples generated in a rigid body physics simulation. Finally, at the grasp selection stage we now optimise across a space defined by this bundle of approach trajectories so as to maximise the chance of reaching a stable grasp. The method copes with partial and noisy shape information for the test objects.

Fig. 1: We want transferrable grasps that are robust to different initial hand-object poses, and thus different interactions during reach to grasp, thus reaching similar final grasp states. We achieve this by learning a set of trajectories that, associated with a model of the final grasp state, form an attractor basin around that state.

A. Related Work

Previous work in learning generalisable grasps falls broadly into two classes. One class of approaches utilises the shape of common object parts or their appearance to generalise grasps across object categories [1], [2], [3], [4]. This works well for low DoF hands. Another class of approaches captures the global properties of the hand shape either at the point of grasping, or during the approach [5]. This global hand shape can additionally be associated with global object shape, allowing generalisation by warping grasps to match warps of global object shape [6]. This second

*This work was supported by EC-FP7-ICT-600918, PacMan.

[1] Marek Kopicki and Jeremy L. Wyatt are with the Intelligent Robotics Laboratory, School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK.

[2] Carlos J. Rosales, Hamal Marino and Marco Gabiccini are with Centro Piaggio, Universita di Pisa, Pisa, Italy.

Correspond to: M.S.Kopicki at cs.bham.ac.uk

Preprint submitted to 2016 IEEE International Conference on Robotics and Automation. Received September 15, 2015.

class works well for high DoF hands, but generalisation is more limited. We have previously achieved the advantages of both classes, generalising grasps across object categories with high DoF hands. In this paper we go beyond this, learning and generalising grasps for under-actuated hands.

Several hands with such behavior have been proposed in the literature with different implementations [7], [8], with a common goal: simplicity plus robustness. Their initial tests under human operation are promising, but autonomous grasping with underactuated hands faces challenges due to the almost non-observability of the finger deformation when the hand is constrained by the environment and/or a target object. Most of the existing planning algorithms for this type of hands boil down to generating good wrist poses and let the adaptive mechanism handle all variation and uncertainty while closing, such as [9], where a sequence of wrist and object poses and the corresponding interaction wrenches are generated, which are expected to exploit environmental constraints. Another approach is that by [11], where static wrist poses are sampled using different strategies around the object from where the fingers are closed using a rigid-body simulator, to finally select the areas of major success rate to generate new wrist poses.

While these approaches exploit, to some extent, the adaptive properties of the underactuated mechanism, they can be improved on. In this paper we show how we can, for the first time, learn grasps for underactuated hands that are then transferred to novel objects. This requires learning representations of the final grasp state that are amenable to transfer to new objects, grouping example grasps by the end grasp state, and learning and optimisation of reach-to-grasp trajectories.

II. OVERVIEW OF APPROACH

In our approach the main steps are as follows. A model of a training object is presented in a rigid body physics simulator. Then a number of example grasps are executed by a human, with the precise motions of hand and object during contact being determined by the simulation. Each example grasp continues until a final stable grasp state is reached. We call this the *equilibrium state*, consisting of the final hand shape, and the final set of contact relations between hand and object. For training and inference purposes each example grasp has two parts: an equilibrium state, and the reach to grasp trajectory leading to it.

We generate the example grasps in sets. Each set corresponds to a type of grasp, e.g. power or pinch. This means that the equilibrium states are similar within a set, but differ substantially between sets.

Models are then learned for each grasp and for each set. Models are learned of the reach to grasp, the hand configuration in the equilibrium state, and the contact relations between hand and object in the equilibrium state. Given these models, when a new object is presented, a (partial) model of that object is obtained by sensing. This model is combined with the models learned from the training grasps.

Then many candidate equilibrium states, and associated candidate reach to grasp trajectories are generated by sampling. Finally they are optimised so as to maximise the likelihood of the grasp according to a product of experts.

III. BASIC REPRESENTATIONS

We now sketch the representations underpinning our approach. We define several models: an object model (partial and acquired from sensing); a model of the contact between a finger link and the object; a model of the whole hand configuration; and a model of the reach to grasp trajectory. First we describe the kernel density representation for all these models. Then we describe the surface features we use to encode some of these models. Then we follow with a description of each model type. Throughout we assume that the robot's hand comprises N_L rigid *links*: a palm, and several phalanges or links. We denote the set of links $L = \{L_i\}$.

A. Kernel Density Estimation

$SO(3)$ denotes the group of rotations in three dimensions. A feature belongs to the space $SE(3) \times \mathbb{R}^2$, where $SE(3) = \mathbb{R}^3 \times SO(3)$ is the group of 3D *poses*, and surface descriptors are composed of two real numbers. We extensively use probability density functions (PDFs) defined on $SE(3) \times \mathbb{R}^2$. We represent these PDFs non-parametrically with a set of K features (or particles) x_j

$$S = \left\{x_j : x_j \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^2\right\}_{j\in[1,K]}. \qquad (1)$$

The probability density in a region is determined by the local density of the particles in that region. The underlying PDF is created through *kernel density estimation* [12], by assigning a kernel function $\mathcal{K}$ to each particle supporting the density, as

$$\mathbf{pdf}(x) \simeq \sum_{j=1}^{K} w_j \mathcal{K}(x|x_j, \sigma), \qquad (2)$$

where $\sigma \in \mathbb{R}^3$ is the kernel bandwidth and $w_j \in \mathbb{R}^+$ is a weight associated to x_j such that $\sum_j w_j = 1$. We use a kernel that factorises into three functions defined by the separation of x into $p \in \mathbb{R}^3$ for position, a quaternion $q \in SO(3)$ for orientation, and $r \in \mathbb{R}^2$ for the surface descriptor. Furthermore, let us denote by μ another feature, and its separation into position, orientation and a surface descriptor. Finally, we denote by σ a triplet of real numbers:

$$x = (p, q, r), \qquad (3a)$$

$$\mu = (\mu_p, \mu_q, \mu_r), \qquad (3b)$$

$$\sigma = (\sigma_p, \sigma_q, \sigma_r). \qquad (3c)$$

We define our kernel as

$$\mathcal{K}(x|\mu, \sigma) = \mathcal{N}_3(p|\mu_p, \sigma_p)\Theta(q|\mu_q, \sigma_q)\mathcal{N}_2(r|\mu_r, \sigma_r) \qquad (4)$$

where μ is the kernel mean point, σ is the kernel bandwidth, $\mathcal{N}_n$ is an n-variate isotropic Gaussian kernel, and Θ corresponds to a pair of antipodal von Mises-Fisher distributions which form a Gaussian-like distribution on $SO(3)$ [13], [14]. The value of Θ is given by

$$\Theta(q|\mu_q, \sigma_q) = C_4(\sigma_q)\frac{e^{\sigma_q\ \mu_q^T q} + e^{-\sigma_q\ \mu_q^T q}}{2} \tag{5}$$

where $C_4(\sigma_q)$ is a normalising constant, and $\mu_q^T q$ denotes the quaternion dot product.

Using this representation, conditional and marginal probabilities can easily be computed from Eq. (2). The marginal density $\mathbf{pdf}(r)$ is computed as

$$\mathbf{pdf}(r) \tag{6}$$

$$= \iint \sum_{j=1}^{K} w_j \mathcal{N}_3(p|p_i, \sigma_p)\Theta(q|q_i, \sigma_q)\mathcal{N}_2(r|r_i, \sigma_r)\mathrm{d}p\mathrm{d}q \tag{7}$$

$$= \sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r), \tag{8}$$

where $x_j = (p_j, q_j, r_j)$. The conditional density $\mathbf{pdf}(p, q|r)$ is given by

$$\mathbf{pdf}(p, q|r) = \frac{\mathbf{pdf}(p, q, r)}{\mathbf{pdf}(r)} \tag{9}$$

$$= \frac{\sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r)\mathcal{N}_3(p|p_j, \sigma_p)\Theta(q|q_j, \sigma_q)}{\sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r)}. \tag{10}$$

B. Surface Features

All objects considered in the paper are represented by point clouds for the purpose of learning and testing. Test object models were constructed from a single view with a depth camera, and were thus incomplete5. We directly augment these points with a frame of reference and a surface feature that is a local curvature descriptor. For compactness, we also denote the pose of a feature as v. As a result,

$$x = (v, r), \qquad v = (p, q). \tag{11}$$

The surface normal at p is computed from the nearest neighbours of p using a PCA-based method (e.g. [15]). The surface descriptors are the local *principal curvatures* [16]. Their directions are denoted $k_1, k_2 \in \mathbb{R}^3$, and the curvatures along k_1 and k_2 form a 2-dimensional feature vector $r = (r_1, r_2) \in \mathbb{R}^2$. The surface normal and the principal directions define the orientation q of a frame that is associated with the point p.

C. Object Model

Thus, given a point cloud, a set of K_O features $\{(v_j, r_j)\}$ can be computed. This set of features defines, in turn, a joint probability distribution, which we call the *object model*:

$$O(v, r) \equiv \mathbf{pdf}^O(v, r) \simeq \sum_{j=1}^{K_O} w_j \mathcal{K}(v, r|x_j, \sigma_x) \tag{12}$$

where O is short for $\mathbf{pdf}^O$, $x_j = (v_j, r_j)$, $\mathcal{K}$ is defined in Eq. (4) with bandwidth $\sigma_x = (\sigma_v, \sigma_r)$, and where all weights are equal, $w_j = 1/K_O$. In summary this object model O represents the object as a pdf over surface normals and curvatures.

IV. LEARNED MODELS

We now describe the representations for each of the three models that are learned from a set of grasp examples. We start with the contact model, proceed with the equilibrium state hand configuration model, and finish with the reach to grasp model.

A. Contact Model

A contact model M_i encodes the joint probability distribution of surface curvatures and of the 3D pose of the i-th hand link in the equilibrium state. Let us consider the hand as having grasped some given training object. The contact model for link L_i is denoted by

$$M_i(U, R) \equiv \mathbf{pdf}_i^M(U, R) \tag{13}$$

where M_i is short for $\mathbf{pdf}_i^M$, R is the random variable modelling surface curvature, and U models the pose of L_i *relative* to the frame of reference defined by the directions of principal curvature and the surface normal. In other words, denoting realisations of R and U by r and u, $M_i(u, r)$ is proportional to the probability of finding L_i at pose u relative to the frame of a nearby object surface patch that exhibits principal curvatures equal to r.

Given a set of features $\{x_j\}_{j=1}^{K_O}$, with $x_j = (v_j, r_j)$ and $v_j = (p_j, q_j)$, a contact model M_i is constructed from them. Features close to the link surface are more important than those lying far from the surface. Features are thus weighted, to make their influence on M_i decrease with their distance to the i^{th} link. We use a weighting function whose value decreases exponentially with the square distance to the link:

$$w_{ij} = \begin{cases} \exp(-\lambda ||p_j - a_{ij}||^2) & \text{if } ||p_j - a_{ij}|| < \delta_i \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where $\lambda \in \mathbb{R}^+$, there is a cut-off distance δ_i, and a_{ij} is the point on the surface of L_i that is closest to p_j. The intuitive motivation for this choice is that we require a weight function that falls off quickly so that the contact model will only take account of the local shape, while falling off smoothly.

Let us denote by $u_{ij} = (p_{ij}, q_{ij})$ the pose of L_i relative to the pose v_j of the j^{th} surface feature. In other words, u_{ij} is defined as

$$u_{ij} = v_j^{-1} \circ s_i, \tag{15}$$

where s_i denotes the pose of L_i, $\circ$ denotes the pose composition operator, and v_j^{-1} is the inverse of v_j, with $v_j^{-1} = (-q_j^{-1}p_j, q_j^{-1})$ (see Fig. 2). The contact model is estimated as

$$M_i(u, r) \simeq \frac{1}{Z}\sum_{j=1}^{K_{M_i}} w_{ij}\mathcal{N}_3(p|p_{ij}, \sigma_p)\Theta(q|q_{ij}, \sigma_q)\mathcal{N}_2(r|r_j, \sigma_r) \tag{16}$$

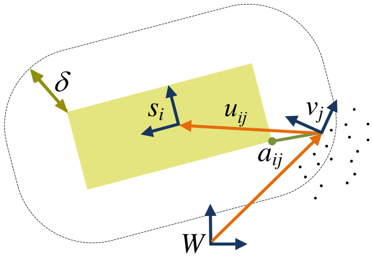

Fig. 2: Contact model. The figure shows the i-th link L_i (solid block) and its pose s_i. The black dots are samples of the surface of an object. The distance a_{ij} between a feature v_j and the closest point on the link's surface is shown. The rounded rectangle illustrates the cut-off distance δ_i. The poses v_j and s_i are expressed in the world frame W. The arrow u_{ij} is the pose of L_i relative to the frame for the surface feature v_j.

where Z is a normalising constant, $u = (p, q)$, and where $K_{M_i} \leq K_O$ is a number of features which are within cut-off distance δ_i to the surface of link L_i. If the number of features K_{M_i} of contact model M_i is not sufficiently large, contact model M_i is not instantiated and is excluded from any further computation. Consequently, the overall number of contact models N_M is usually smaller than the number of links N_L of the robotic hand. We denote the set of contact models learned from a grasp example g as $\mathcal{M}^g = \{\mathcal{M}_i^g\}$.

The parameters λ and σ_p, σ_q, σ_r were chosen empirically. The time complexity for learning each contact model from an example grasp is $\Omega(TK_O)$ where T is the number of triangles in the tri-mesh describing the hand links, and K_O is the number of points in the object model.

B. Equilibrium State Hand Configuration Model

The equilibrium state hand configuration model, denoted $h_c^e(j) \in \mathbb{R}^D$, for the grasp examples $j = 1 \dots k$ within the set of k training examples of a particular grasp type g. The purpose of this model is to restrict the grasp search space (during grasp transfer) to hand configurations that resemble those observed during training. We combine the configurations for the examples $j = 1 \dots k$ to create a single mixture model density:

$$C^g(h_c^e) \equiv \sum_{j=1}^{k} \mathcal{N}_D(h_c^e | h_c^e(j), \sigma_{h_c^e}) \qquad (17)$$

This expresses a density over hand configurations in the equilibrium state for a grasp type g.

C. Reach to Grasp Model

For a particular grasp type, in addition to modelling the equilibrium states of the hand, we must also model the trajectories taken to reach those equilibrium states. A single reach to grasp trajectory for an underactuated hand has three elements: the tool centre point (wrist) trajectory, the hand configuration trajectory, and the motor signal trajectory. We assume that a trajectory starts at time t_0 and ends in the equilibrium state at time t_e. We denote the wrist trajectory $h_w^{0:e}$, the hand configuration trajectory $h_c^{0:e}$, and the motor signal trajectory $h_m^{0:e}$ respectively. The motor signal can be a wide variety of signals in practice. Here we choose it to be the position of the single actuator. When the hand is not in contact with an object the motor signal and the wrist pose together determine the hand configuration. When in contact, the actual hand configuration will differ. The reach to grasp model is simply the concatenation of each component $(h_w^{0:e}, h_c^{0:e}, h_m^{0:e})$. The set of reach to grasp trajectory models define an attractor basin, leading towards the final hand configuration.

In the next section we explain how we gather the grasp examples that are used to learn these models. Then in Section VI the inference method—by which the models are used to generate grasps for new objects—is described.

V. DATA GENERATION

There are several ways to implement underactuation in a dexterous hand. In this paper we employ an approach based on adaptive synergy transmission, due to its simplicity and robust design, and its ability for complex interaction with the environment. The Pisa/IIT SoftHand [7] implements such a transmission mechanism. This hand has 19 degrees of freedom (DoF) distributed over four fingers and an opposable thumb, but only 1 degree of actuation (DoA). The synergy motion of the hand in free space has been derived from databases of human hand postures. The overall behaviour parameters are the matrices that correspond to the transmission ratio, R, to the joint stiffness, K_q. The actuation is done through a single tendon routed throuh all joints, making the fingers flex and abduct.

Moving such a hand to grasp an object results in a hard-to-predict contact and hand shapes due to the adaptivity. We thus generate a variety of grasp examples to cover a portion of the interaction space. However, recording many trajectories of all the finger elements that affect the grasp in the real world is non-trivial. For this reason, we generate the example interactions for training using a rigid-body physics simulator, where these problems are avoided. The main two simulation elements we have developed are the contact stability model and the hand behavior model. In the case of the Pisa/IIT softhand, the latter depends heavily on the former. We used the standard distribution of Gazebo and Open Dynamic Engine, both in widespread use. The adaptive synergy equations have been implemented as a plugin to these, and accompany the proper kinematic description of the Pisa/IIT SoftHand[1].

At the current state, there are no generally accepted measures concerning whether a grasp by an underactuated hand is good or not, hence the lack of robust grasp planners for them is not a surprise. Thus, generating a large dataset at this point is useless, and there are plans in the future to cover this area. As a result, we generated the examples by

[1] The Pisa/IIT SoftHand ROS/Gazebo packages are available at `https://github.com/CentroEPiaggio/pisa-iit-soft-hand`

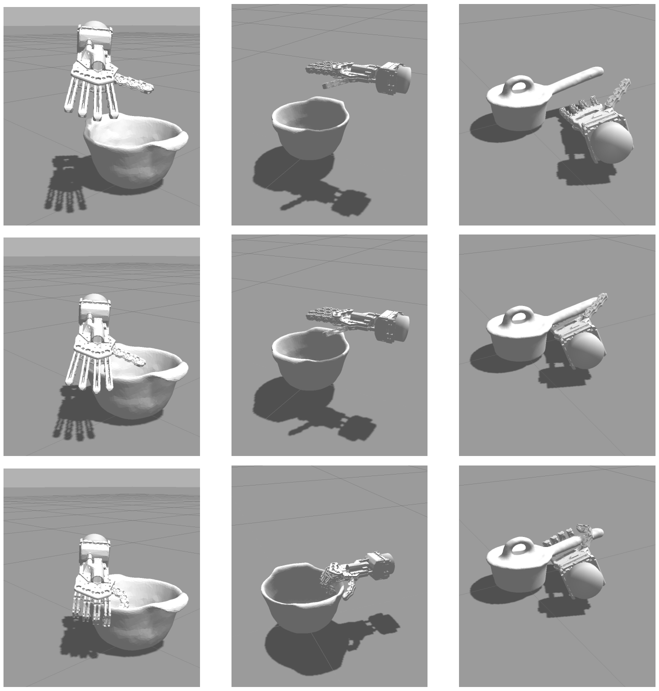

Fig. 3: Snapshots of the simulation of a pinch and rim grasp types for the colander (first and second columns), and handle grasp for the pot (last column).

guiding manually the hand to a “nice” grasp as shown in Fig. 3. In this simpler scenario, we assume without loss of generality that the grasps are labelled by type. In our example dataset, we have three grasp types namely pinch, rim and by-the-handle. The main difference between pinch and rim is the fingers configuration w.r.t. the top border of objects. In the pinch grasp, the thumb goes inside whereas in the rim grasp, the fingers go inside. In the latter, the container can be filled with liquid while holding, for instance. For each grasp example, the corresponding dataset comprises the set of trajectories as described in the previous section.

VI. INFERENCE

After acquiring the models from a set of training grasps, we present the robot with a test (query) object. The goal is to find a generalisation of the training grasp that is likely according to all of the model types simultaneously. First of all, we obtain a point cloud for the test object, and thus an object model. We then combine every contact model with that object model, so as to obtain a set of *query densities*, one for each link with a contact model defined for the example grasp. The i-th query density Q_i is a density modelling where the i-th link can be placed, in the equilibrium state, with respect to the surface of a new object.

From the query densities, a candidate equilibrium grasp state is generated. This is then augmented with a reach to grasp trajectory that finishes close to the candidate equilibrium grasp state. Finally we refine the equilibrium grasp and reach to grasp by performing a simulated annealing search in the space of equilibrium state wrist poses and hand configurations, so as to maximise the grasp likelihood. We repeat the entire process many times. This procedure generates many possible grasps, ranked by likelihood. We give details below.

Algorithm 1: Pose sampling (M_i, O)

```
For samples $j = 1$ to $K_{Q_i}$
    Sample $(\hat{v}_j, \hat{r}_j) \sim O(v, r)$
    Sample from conditional density $(\hat{u}_{ij}) \sim M_i(u|\hat{r}_j)$
    Compute sample weight $w_{ij} = M_i(\hat{r}_j)$
    $\hat{s}_{ij} = \hat{v}_j \circ \hat{u}_{ij}$
    separate $\hat{s}_{ij}$ into position $\hat{p}_{ij}$ and quaternion $\hat{q}_{ij}$
return $\{(\hat{p}_{ij}, \hat{q}_{ij}, w_{ij})\}, \forall j$
```

A. Query Density

A query density is, for a hand link and an object model, a density over the pose of that hand link relative to the object. Intuitively the query density encourages a finger link to make contact with the object at locations that have similar local surface curvature to that in the training example. Specifically, we use K_{Q_i} kernels centred on the set of weighted finger link poses:

$$Q_i(s) \simeq \sum_{j=1}^{K_{Q_i}} w_{ij} \mathcal{N}_3(p|\hat{p}_{ij}, \sigma_p) \Theta(q|\hat{q}_{ij}, \sigma_q) \qquad (18)$$

with j-th kernel centre $(\hat{p}_{ij}, \hat{q}_{ij}) = \hat{s}_{ij}$, and weights are normalised $\sum_j w_{ij} = 1$. When a test object is presented, a set of query densities $\mathcal{Q}^g$ is calculated for the equilibrium state for each training grasp b for the grasp type g. The set $\mathcal{Q}^g_b = \{Q^g_{b,i}\}$ has $N^g_Q = N^g_M$ members, one for each contact model M^g_i in $\mathcal{M}^g$. We estimate the query density using a Monte Carlo procedure detailed in Alg.1.

B. Equilibrium Grasp Generation

Once query densities have been created for the new object for each training example, an initial set of equilibrium state grasps is generated for each grasp type g. For each candidate equilibrium grasp of a particular grasp type we proceed as follows. First an example grasp is selected at random. Then a finger link is selected at random. This ‘seed’ link indexes its query density Q^g_i. A link pose s_i is then sampled from that query density. Then an equilibrium state hand configuration h^e_c is sampled from C^g. Together the seed link and the hand configuration define a complete equilibrium state hand pose h in the workspace via forward kinematics. This is an initial ‘seed’ grasp, which will subsequently be refined. A large set of such initial solutions is generated, where $h^g_e(j) = (h^e_w(j), h^e_c(j))$ means the j^{th} initial solution for grasp type g.

C. Reach to Grasp Generation

Given an equilibrium grasp, a reach to grasp trajectory is selected and adapted to maximise the chance of reaching that equilibrium grasp state. Specifically, we sample a reach to grasp model $(h^{0:e}_w, h^{0:e}_c, h^{0:e}_m)$ according to a multinomial

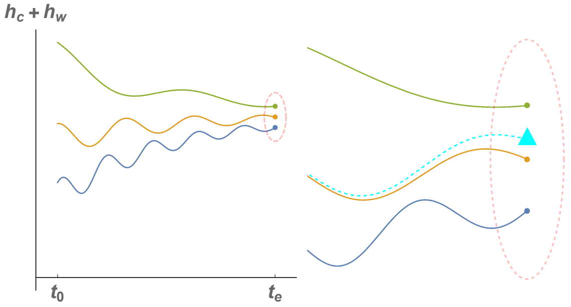

Fig. 4: (Left) We acquire set of reach to grasp trajectories that, associated with a model of the final equilibrium grasp state, form an attractor basin around that state. (Right) When an equilibrium state hand configuration is generated (triangle) a reach to grasp trajectory is sampled, and then the configuration component of the trajectory is smoothly interpolated between the selected reach to grasp and the generated equilibrium state grasp.

probability distribution created by the normalised values of a Gaussian centred on the candidate equilibrium grasp $h^g_e(j)$. To align the selected reach to grasp to the candidate equilibrium grasp, the wrist trajectory $h^{0:e}_w$ is trivially redefined to be relative to the frame $h^e_w(j)$. Then the configuration trajectory only $h^{0:e}_c$ is warped (see Fig. 4) so that it smoothly shifts from h^0_c to $h^e_c(j)$ from the beginning to the end of the trajectory. Having generated an initial solution set $\mathcal{H}^1$ stages of optimisation and selection are then interleaved.

D. Grasp Optimisation

The objective of the grasp optimisation steps is, given a candidate equilibrium grasp and a reach to grasp model, to find a grasp that maximises the product of the likelihoods of the query densities and the hand configuration density

$$\operatorname*{argmax}_{(h)} \mathcal{L}^g(h) \tag{19}$$

$$= \operatorname*{argmax}_{(h)} \mathcal{L}^g_C(h)\mathcal{L}^g_Q(h) \tag{20}$$

$$= \operatorname*{argmax}_{(h_w,h_c)} C^g(h_c) \prod_{Q^g_i \in \mathcal{Q}^g} Q^g_i \left(k^{\text{for}}_i (h_w, h_c)\right) \tag{21}$$

where $\mathcal{L}^g(h)$ is the overall likelihood, where $C^g(h_c)$ is the hand configuration model (17), Q^g_i are query densities (18). Thus whereas each initial grasp is generated using only a single query density, grasp optimisation requires evaluation of the grasp against all query densities. It is only in this improvement phase that all query densities must be used. Improvement is by simulated annealing (SA) [17]. The SA temperature T is declined linearly from T_1 to T_K over the K steps. In each time step, one step of simulated annealing is applied to every grasp m in $\mathcal{H}^k$.

E. Grasp Selection

At predetermined selection steps (here steps 1 and 50 of annealing), grasps are ranked and only the most likely 10% retained for further optimisation. During these selection steps the criterion in (21) is augmented with an additional expert $W(h_w, h_c)$ penalising collisions for the entire reach to grasp trajectory in a soft manner. This soft collision expert has a cost that rises exponentially with the greatest degree of penetration through the object point cloud by any of the hand links. We thus refine Eq. 21:

$$\begin{aligned} \mathcal{L}^g(h) &= \mathcal{L}^g_W(h)\mathcal{L}^g_C(h)\mathcal{L}^g_Q(h) && (22) \\ &= W(h_w, h_c) C^g(h_c) \prod_{Q_i \in \mathcal{Q}^g} Q^g_i \left(k^{\text{for}}_i (h_w, h_c)\right) && (23) \end{aligned}$$

where $\mathcal{L}^g(h)$ is now factorised into three parts, which evaluate the collision, hand configuration and query density experts, all at a given hand pose h. A final refinement of the selection criterion is due to the fact the number of links involved in a grasp varies across grasp types. Thus the number of query densities $N^{g_1}_Q$, $N^{g_2}_Q$ for different grasp models $g_1 \neq g_2$ also varies, and so the values of $\mathcal{L}^{g_1}$ and $\mathcal{L}^{g_2}$ cannot be compared directly. Given the grasp with the maximum number of involved links $N^{\max}_Q$, we therefore normalise the likelihood value (22) with

$$\|\mathcal{L}^g(h)\| = \mathcal{L}^g_W(h)\mathcal{L}^g_C(h)\left(\mathcal{L}^g_Q(h)\right)^{\frac{N^{\max}_Q}{N^g_Q}} . \tag{24}$$

It is this normalised likelihood $\|\mathcal{L}^g\|$ that is used to rank all the generated grasps across all the grasp types during selection steps. After simulated annealing has yielded a ranked list of optimised grasp poses, they are checked for reachability given other objects in the workspace, and unreachable poses are pruned.

F. Grasp Execution

The remaining best scoring hand pose $h*$ is then used to generate a reach to grasp trajectory. Since the hand is underactuated this consists of the wrist pose trajectory, and the motor signal trajectory. This is the command sequence that is executed on the robot.

VII. Results

The experiments were conducted as follows. Training consisted of nine example grasps, executed in simulation, with a human in control. These nine grasps were grouped into three grasp types (rim, pinch, and handle). The rim and pinch grasp types were trained on the colander object, and the handle grasp type was demonstrated on the saucepan. During testing an object was placed on the table. Every grasp type was compared automatically, and one selected for execution according to the methods described above. The models of the test objects consisted of a point cloud taken from just one view. Thus reconstructions were partial, typically less that 25% of the object's surface area. No test objects had been seen previously by the robot, and it can be seen from Fig. 5. Fifteen test objects were presented, and 12 of the 15 test grasps succeeded, giving a generalisation success rate of 80%. While the difference is not statistically significant, this is slightly higher than the 77.7% success rate we recorded

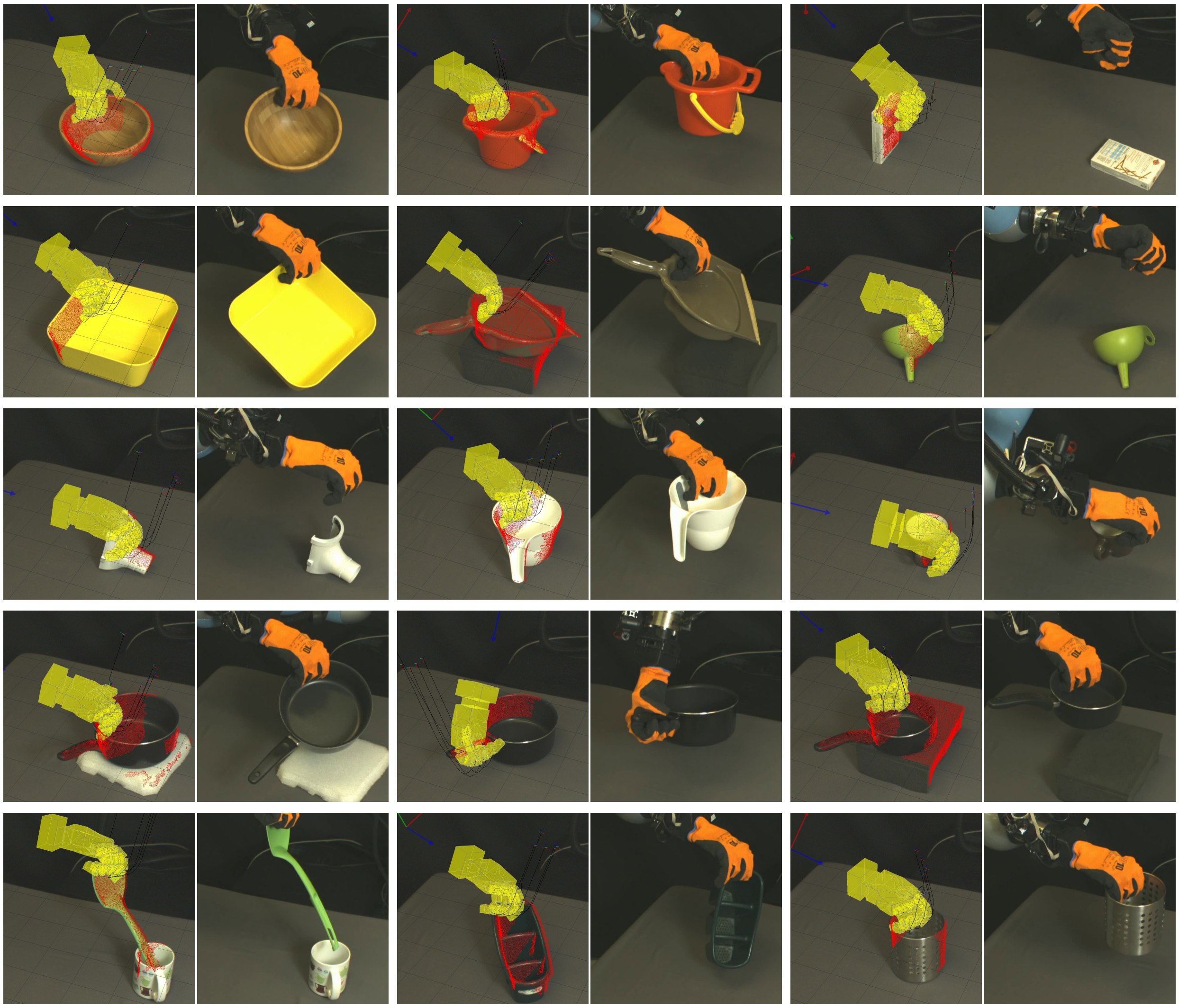

Fig. 6: The fifteen test grasps. Each one has a pair of images. The predicted equilibrium grasp state is shown on the left of each pair, and the actual grasp on the right. Counting from top left it can be seen that grasps 3,6, and 7 failed. All other grasps succeeded.

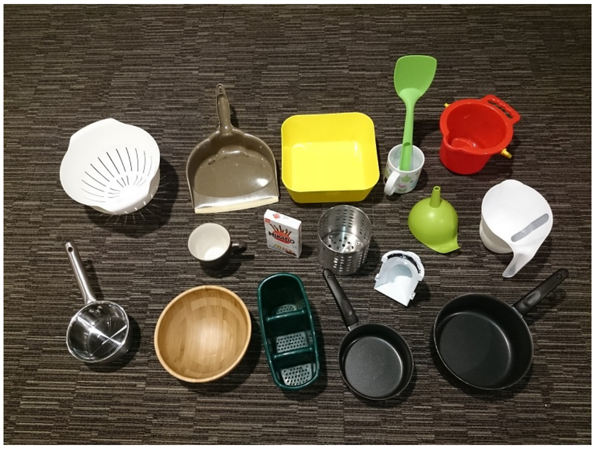

Fig. 5: The two training objects are on the far left. The testing objects on the right. 12 from 15 test grasps on novel objects were successful.

on a larger test set for a fully actuated hand also working from a single view of an object [18].

Finally, it is worth noting the similiarity between the location of the actual and targetted final grasp states. In a majority of cases the grasp involved interactions with the object, moving it to the stable grasp pose. This is a natural property of the hand, but it might have been supposed that the learning method would not be robust to such interactions in terms of the accuracy of the grasp.

References

[1] A. Saxena, L. Wong, and A. Ng, "Learning grasp strategies with partial shape information," in *Proceedings of AAAI*. AAAI, 2008, pp. 1491–1494.

[2] R. Detry, C. H. Ek, M. Madry, and D. Kragic, "Learning a dictionary of prototypical grasp-predicting parts from grasping experience," in *International Conference on Robotics and Automation*. IEEE, 2013, pp. 601–608.

[3] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, “Learning of grasp selection based on shape-templates,” *Autonomous Robots*, vol. 36, no. 1-2, pp. 51–65, 2014.

[4] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, “A kernel-based approach to direct action perception,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2605–2610.

[5] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, “Generalization of human grasping for multi-fingered robot hands,” in *International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2043–2050.

[6] U. Hillenbrand and M. Roa, “Transferring functional grasps through contact warping and local replanning,” in *IEEE/RSJ International Conference on Robotics and Systems*. IEEE, 2012, pp. 2963–2970.

[7] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, “Adaptive synergies for the design and control of the Pisa/IIT SoftHand,” *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.

[8] A. M. Dollar and R. D. Howe, “The highly adaptive SDM hand: Design and performance evaluation,” *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 585–597, 2010.

[9] C. Eppner and O. Brock, “Planning grasp strategies that exploit environmental constraints,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2015.

[10] M. Bonilla, E. Farnioli, C. Piazza, M. G. Catalano, G. Grioli, M. Garabini, M. Gabiccini, and A. Bicchi, “Grasping with soft hands,” in *In Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2014.

[11] M. Bonilla, D. Resaco, M. Gabiccini, and A. Bicchi, “Grasp planning with soft hands using bounding box object decomposition,” in *In Proceedings of the IEEE International Conference of Intelligent Robots and Systems*, 2015.

[12] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.

[13] R. A. Fisher, “Dispersion on a sphere,” in *Proc. Roy. Soc. London Ser. A.*, vol. 217, no. 1130. Royal Society, 1953, pp. 295–305.

[14] E. B. Sudderth, “Graphical models for visual object recognition and tracking,” Ph.D. dissertation, MIT, Cambridge, MA, 2006.

[15] K. Kanatani, *Statistical optimization for geometric computation: theory and practice*. Courier Dover Publications, 2005.

[16] M. Spivak, *A comprehensive introduction to differential geometry*. Publish or Perish Berkeley, 1999, vol. 1.

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[18] M. Kopicki, R. Detry, M. Adjigble, A. Leonardis, and J. L. Wyatt, “One shot learning and generation of dexterous grasps for novel objects,” *International Journal of Robotics Research*, 2015.