

# Learning Dexterous Grasps That Generalise To Novel Objects By Combining Hand And Contact Models

Marek Kopicki, Renaud Detry, Florian Schmidt, Christoph Borst, Rustam Stolkin, and Jeremy L. Wyatt

**Abstract**—Generalising dexterous grasps to novel objects is an open problem. We show how to learn grasps for high DoF hands that generalise to novel objects, given as little as one demonstrated grasp. During grasp learning two types of probability density are learned that model the demonstrated grasp. The first density type (the contact model) models the relationship of an individual finger part to local surface features at its contact point. The second density type (the hand configuration model) models the whole hand configuration during the approach to grasp. When presented with a new object, many candidate grasps are generated, and a kinematically feasible grasp is selected that maximises the product of these densities. We demonstrate 31 successful grasps on novel objects (an 86% success rate), transferred from 16 training grasps. The method enables: transfer of dexterous grasps within object categories; across object categories; to and from objects where there is no complete model of the object available; and using two different dexterous hands.

## I. INTRODUCTION

Transferring dexterous grasps to novel objects is an open problem. We present a method that infers grasps for novel objects from as little as one training example. It generalises a human chosen *grasp type* (e.g. rim, pinch) trained from one or two examples, to a novel object of potentially quite different shape. We demonstrate learning and transfer of six grasps types to 31 different grasps of 18 objects, and for two different dexterous hands. The results show the ability to generalise a grasp within and across object categories, and with full or partial shape information. The method requires no knowledge of the human defined object category (e.g. cup, box) when learning or performing transfer.

There has been progress in learning generalisable grasps. One class of approaches utilises the shape of common object parts to generalise grasps across object categories [8], [11], [15]. 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 [2]. This global hand shape can additionally be associated with global object shape, allowing generalisation by warping grasps to match warps of global object shape [12]. This second class works well for high DoF hands, but



Fig. 1. Objects used: the four objects on the left were used solely as example grasps, the ten objects on the right were solely used as novel test objects. The four at the back were used as either training or test, but not for both at the same time, e.g. we transferred a pinch grasp from the coke bottle to the vitamin tube, and a pinch support grasp vice versa.

generalisation is more limited. We achieve the advantages of both classes, generalising grasps across object categories with high DoF hands.

Our main technical innovation to achieve this is to learn two types of models from the example grasp, and then recombine them when inferring a new grasp. Both models are probability density functions. The first is a *contact model* of the relation between a finger part and the local object shape at its contact. We learn one such model for each rigid link in the hand. Each model can generate many possible independent placements for its link on a new “query” object. Given these models the next problem is to find a hand configuration that combines a good subset of placements in a kinematically feasible grasp. To address this we learn a second type of model, a *hand configuration model* from the example grasp. We then use this hand configuration model to constrain the combined search space for the link placements.

Given these two learned models we can infer grasps for novel objects. When presented with a novel query object we use a Monte Carlo procedure to construct a third type of density function (called the *query density*) over possible contact points on the new object. We again construct a query density for every link of the hand. We then use these densities in two ways to select the new grasp. First we pick a link, and draw a contact point for it on the object from the query density. Then we sample a hand configuration to obtain the complete set of link poses. The whole solution is then refined using simulated annealing on the product of all the query densities and the hand configuration density. The paper starts with a survey of related work (Sec II); continues with the representations employed and the learning process (Sec III); followed by a description of the process for finding a grasp

Marek Kopicki, Rustam Stolkin and Jeremy L. Wyatt are with the University of Birmingham, England. Email: {msk, stolkinr, jlw}@cs.bham.ac.uk. Renaud Detry is with the EECS department, University of Liege, Belgium. Email: renaud.detry@ulg.ac.be. Florian Schmidt and Christoph Borst are with the German Aerospace Center (DLR), Wessling, Germany, Email: {florian.schmidt, christoph.borst}@dlr.de. The work described here is protected by Patent Application (UKIPO 1309156.6). We gratefully acknowledge support by the FP7-IST-600918 (PaCMan), FP7-IST-248273 (GeRT), the Swedish Research Council (VR), the Belgian National Fund for Scientific Research (FNRS).

for a novel object. We finish with an experimental study (Sec VI) and a discussion.

## II. RELATED WORK

In robotics, mainstream grasp planning has traditionally relied on force analysis [3]. Unfortunately, the strengths of force analysis become mitigated in scenarios where the object models that the robot can recover (from memory or sensing) are incomplete or erroneous. Some authors have explored ways of directly linking perceptions to action parameters. Symmetry principles have been used to reconstruct the occluded side of unknown objects [23], and thereby allow the definition of grasp contact points on occluded surfaces [4]. Others have parameterised grasps by looking for shapes that are likely to fit into the robot's gripper [9], [17]. Popovic et al. [17] computed grasps applied to object edges detected in 2D images. Another class of methods search 3D range data for shapes closely matching the geometry of the gripper [9].

Instead of hard-coding the function that computes grasp parameters from vision, a growing number of researchers have focused on methods that learn the perception-action mapping from experimental data [7], [16]. A subclass of these focused on learning a mapping from incomplete object views to grasp parameters [1], [8], [11], [15], [14], [19], while another subclass has aimed at transferring grasps across objects whose complete 3D shape is known [2], [12]. When addressing the problem of grasping a partially-perceived object, authors have developed means of learning how to place the wrist of the gripper with respect to the part of the object that is perceived by the robot's vision system [8], [11], [15], [14]. When working with two-finger grippers, the wrist completely parametrises the gripper pose [11]. In order to plan grasps for a three-finger hand, Kroemer et al. [15] and Detry et al. [8] let the robot learn a hand preshape, and relied on compliance or force sensing to close the hand. Alternately, Kroemer et al. [14] also relied on control policies to adapt the fingers to the visual input. Saxena et al. [19] learned a three-finger grasp success classifier from a bank of photometric and volumetric object features.

Hillenbrand et al. [12] addressed the problem of transferring a multi-finger grasp between two objects of known 3D shape. A known object's geometry is warped until it matches that of a novel object, thereby also warping grasp points on the surface of the known object onto candidate grasp points on the novel object. Ben Amor et al. [2] exploit this warping method to transfer grasps taught by a human hand (using a data glove) to contact points for a robot hand on a novel object. Our work is, in spirit, most similar to the work of Ben Amor et al. [2] and Hillenbrand et al. [12], and to the work of Saxena et al. [19]. We compute a full hand grasping configuration for a novel object, using a grasp model that is learned from a single or a few example grasps. Our method performs best when a full shape model of the target object is available, but it is also applicable to partially-modelled objects. One difference with [12] is that they transfer grasps within the same object shape category, whereas we are able to transfer grasps to different object categories.

## III. REPRESENTATIONS

This paper addresses the problem of transferring grasps across objects. Given one or more grasp examples, we aim to capture the configuration of the hand and its fingers with respect to neighbouring object surfaces, and to adapt the hand and fingers to the shape of a novel object.

We assume that the robot's hand is formed of a palm, and a number of finger phalanges. We refer to the palm and phalanges as *parts* of the hand, denoted by  $H_i$  ( $i = 1, \dots, N$ ). Our approach relies on a pair of models that capture two complementary aspects of the grasping process. The first model, referred to as the *contact model*, captures the configuration of one hand part  $H_i$  relative to local features from the object's surface. Given the 3D model of an object that the robot has not grasped yet, contact model  $M_i$  allows the robot to compute a set of poses for the hand part  $H_i$ . When a robot is equipped with a  $N$ -part hand, up to  $N$  contact models can be instantiated.

The second model, referred to as the *hand configuration model*, captures a set of whole hand shapes close to the whole hand shape in the example grasp. The role of the hand configuration model in inferring a grasp for a novel object will be to rapidly constrain the search space for the hand part positions. The hand configuration model is learned from the same grasp data as the contact models.

### A. Kernel Density Estimation

Much of our work relies on the joint probabilistic modeling of  $SE(3)$  poses and surface descriptors. As a result, this paper makes extensive use of probability density functions (PDFs) defined on  $\mathbb{R}^3 \times SO(3) \times \mathbb{R}^2$ , where  $SO(3)$  is the group of 3D rotations, and surface descriptors are parametrized by two real numbers. This section explains how we define these density functions.

We represent PDFs non-parametrically with a set of pose-descriptor pairs (or particles)  $x_\ell$

$$\{x_\ell : x_\ell \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^2\}_{\ell \in [1, L]}. \quad (1)$$

The probability density in a region of space is determined by the local density of the particles in that region. The underlying PDF is created through *kernel density estimation* [20], by assigning a kernel function  $\mathcal{K}$  to each particle supporting the density, as

$$\text{pdf}(x) \simeq \sum_{\ell=1}^L w_\ell \mathcal{K}(x|\mu_\ell, \sigma_\ell), \quad (2)$$

where  $\mu_\ell$  is the kernel mean point,  $\sigma_\ell$  is the kernel bandwidth and  $w_\ell$  is a weight associated to  $x_\ell$  such that  $\sum_\ell w_\ell = 1$ . We use a kernel that factorises into three functions defined on the three components of our domain, namely  $\mathbb{R}^3$ ,  $SO(3)$ , and  $\mathbb{R}^2$ . Denoting the separation of  $x$  into a position, orientation, and feature by  $(p, q, r)$ , we define our kernel with

$$\mathcal{K}(x|\mu, \sigma) = \mathcal{N}_3(p|\mu_p, \sigma_p) \Theta(q|\mu_q, \sigma_q) \mathcal{N}_2(r|\mu_r, \sigma_r) \quad (3)$$

with kernel mean point  $\mu = (\mu_p, \mu_q, \mu_r)$  and kernel bandwidth  $\sigma = (\sigma_p, \sigma_q, \sigma_r)$ , and where  $\mathcal{N}_k$  is  $k$ -variate isotropic

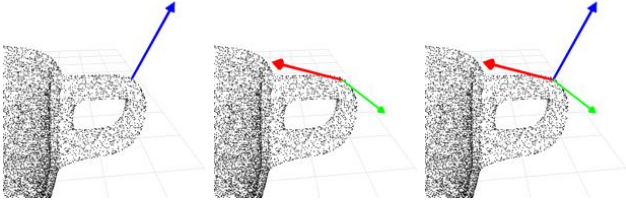


Fig. 2. An example object point cloud (black dots) with a selected point, its surface normal (blue axis), 1-st highest principal curvature (red axis), second lowest principal curvature (green axis), composing a complete frame shown in the right panel.

Gaussian kernel,  $\Theta$  corresponds to a pair of antipodal von-Mises Fisher distributions which forms a Gaussian-like distribution on  $SO(3)$  (for details see [10], [22]). The value of  $\Theta$  is given by

$$\Theta(q|\mu_q, \sigma_q) = C_4(\sigma_q) \frac{e^{\sigma_q \mu_q^T q} + e^{-\sigma_q \mu_q^T q}}{2} \quad (4)$$

where  $C_4(\sigma_q)$  is a normalising constant.

### B. Surface Features

All objects considered in the paper are represented by point clouds. Each point  $x$  is characterised by a *feature*, comprising a 3D position  $p$ , a 3D orientation  $q$ , and a surface descriptor  $r$ :

$$x = (v, r), \quad v = (p, q) \quad (5)$$

The surface descriptor corresponds to the local *principal curvatures* [21]. Curvatures at point  $p$  are recorded along two directions that both lie in the plane tangential to the object's surface, i.e. perpendicular to the surface normal at  $p$ . The first direction,  $k_1$ , is the direction of highest curvature. The second direction,  $k_2$ , is perpendicular to  $k_1$ . The curvatures along  $k_1$  and  $k_2$  are denoted by  $r_1$  and  $r_2$  respectively, forming a 2-dimensional feature vector  $r = (r_1, r_2)$ . The surface normals and curvatures are computed using the PCL library [18].

The surface normals and curvatures allow us to define a reference frame  $v$  at each object point  $p$ , with the exception of  $r_1 \approx 0$ , when the principal directions are chosen randomly, as it is in the case of planar surfaces. Fig. 2 illustrates surface normals and curvatures.

The procedure described above allows the computation of a set of features (pose-descriptor pairs)  $\{(v_\ell, r_\ell)\}$  from a given object point cloud. In turn, the set of features defines a joint probability distribution, further referred to as the *object model*:

$$O(v, r) \equiv \text{pdf}^O(v, r) \simeq \sum_{\ell=1}^L w_\ell \mathcal{K}(v, r | \mu_{x_\ell}, \sigma_x) \quad (6)$$

where  $O$  is short for  $\text{pdf}^O$ ,  $x_\ell = (v_\ell, r_\ell)$ , kernels  $\mathcal{K}$  are defined in Eq. (3), and all weights are equal  $w_\ell = 1/L$ .

### C. Contact Model

A contact model  $M_i$  encodes the joint probability distribution of surface features and the 3D pose of the  $i^{\text{th}}$  hand part  $H_i$ . Let us consider the hand grasping some given object. The (object) contact model of hand part  $H_i$  is denoted by

$$M_i(U, R) \equiv \text{pdf}_i^M(U, R) \quad (7)$$

where  $M_i$  is short for  $\text{pdf}_i^M$ ,  $R$  is the random variable modeling surface features, and  $U$  models the pose of  $H_i$  relative to a surface feature. In other words, denoting realisations of  $R$  and  $U$  by  $r$  and  $u$ ,  $M_i(u, r)$  is proportional to the probability of finding  $H_i$  at pose  $u$  relative to the frame of a nearby object surface feature that exhibits feature vector equal to  $r$ .

Given a set of surface features  $\{(v_\ell, r_\ell)\}$ , a contact model  $M_i$  is constructed by selecting a set of features from the neighbourhood of the hand part  $H_i$  (Fig. 3). We proceed with a soft selection which associates, to each surface feature  $x_\ell = ((p_\ell, q_\ell), r_\ell)$ , a weight that exponentially decreases with a squared distance to  $H_i$  (Fig. 4). The weight is given by

$$w_{i\ell} = \exp(-\lambda \|p_\ell - a_{i\ell}\|^2), \quad (8)$$

where  $\lambda \in \mathbb{R}^+$  and  $a_{i\ell}$  is a point on the surface of  $H_i$ , the closest to  $p_\ell$ .

The contact model is approximated as

$$M_i(u, r) \simeq \sum_{\ell} w_{i\ell} \mathcal{N}_3(p | \mu_{p_{i\ell}}, \sigma_{p_i}) \Theta(q | \mu_{q_{i\ell}}, \sigma_{q_i}) \mathcal{N}_2(r | \mu_{r_\ell}, \sigma_r) \quad (9)$$

where all weights are normalised  $\sum_{\ell} w_{i\ell} = 1$ ,  $u = (p, q)$ , and where  $(\mu_{p_{i\ell}}, \mu_{q_{i\ell}}) = u_{i\ell}$  is the centre of a kernel computed from  $\ell$ -th feature pose  $v_\ell$  and the  $i$ -th hand part pose  $s_i$  from their mutual geometric relation (see Fig. 3)

$$s_i = v_\ell \circ u_{i\ell} \quad (10)$$

or equivalently  $u_{i\ell} = v_\ell^{-1} \circ s_i$ , multiplying both sides of (10) by  $v_\ell^{-1}$ .  $u_{i\ell}$  represents frame  $s_i$  of hand part  $H_i$  relative to feature frame  $v_\ell$  and *independently* on the world frame encoded by  $s_i$  itself.

Sum 9 involves only terms for which  $x_\ell = ((p_\ell, q_\ell), r_\ell)$  belong to the neighbourhood of  $H_i$ ,  $\{x_\ell : \|p_\ell - a_{i\ell}\| \leq \delta, \delta \in \mathbb{R}^+\}$ . If the neighbourhood of a particular hand part  $H_i$  is empty, the corresponding contact model is not instantiated and it is excluded from any further computation.

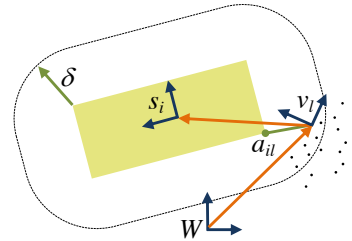


Fig. 3. Contact model:  $i$ -th hand part (solid rectangle) with frame  $s_i$ , its neighbourhood (rounded dashed contour at distance  $\delta$  from the hand part), point cloud (dots), world frame  $W$ ,  $\ell$ -th feature with frame  $v_\ell$  and the corresponding closest point on the part surface  $a_{i\ell}$ .

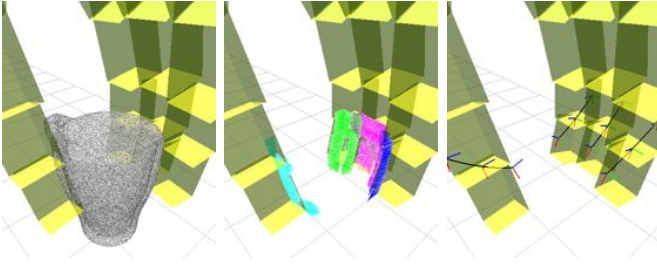


Fig. 4. Example top grasp of a mug represented by a point cloud (left). The colorful regions at contacts between fingers and the object are rays between features and the closest hand part surfaces (middle). The black curves with frames at the fingertips represent the range of hand configurations in Eq. (12) (right).

$\lambda$  in (8) and the bandwidths  $\sigma_{p_i}$ ,  $\sigma_{q_i}$ ,  $\sigma_r$  in (9) were hand tuned and kept fixed in all the experiments. The time complexity for learning each contact model from an example grasp is  $\Omega(|r||O||Tr|)$  where  $|Tr|$  is the number of triangles in the tri-mesh describing the hand part,  $|O|$  is the number of points in the object model.

#### D. Hand Configuration Model

The hand configuration model, denoted by  $C$ , encodes a set of configurations of the hand joints  $h \in \mathbb{R}^D$  (i.e. joint angles), that are particular to example grasps. The purpose of this model is to allow us to restrict the grasp search space (during grasp transfer) to hand configurations that resemble those observed while training the grasp.

In order to boost the generalisation capability of the grasping algorithm the hand configuration model encodes the hand configuration that was observed when grasping the training object, but also a set of configurations recorded during the approach towards the object. Let us denote by  $h^t$  the joint angles at some small distance *before* the hand reached the training object, and by  $h^g$  the hand joint angles at the time when the hand made contact with the training object. We consider a set of configurations interpolated between  $h^t$  and  $h^g$ , and extrapolated beyond  $h^g$ , as

$$h(\gamma) = (1 - \gamma)h^g + \gamma h^t \quad (11)$$

where  $\gamma \in \mathbb{R}$ . For all  $\gamma < 0$ , configurations  $h(\gamma)$  are beyond  $h^g$  (see Fig. 4). The hand configuration model  $C$  is constructed by applying kernel density estimation to  $\{h(\gamma) : \gamma \in [-\beta, \beta], \beta \in \mathbb{R}^+\}$ :

$$C(h) \equiv \sum_{\gamma \in [-\beta, \beta]} w(h(\gamma)) \mathcal{N}_D(h|h(\gamma), \sigma_h) \quad (12)$$

where  $w(h(\gamma)) = \exp(-\alpha \|h(\gamma) - h^g\|^2)$  and  $\alpha \in \mathbb{R}^+$ .  $\alpha$  and  $\beta$  were hand tuned and kept fixed in all the experiments. The hand configuration model computation has time complexity  $\Omega(|\Gamma|)$  where  $\Gamma$  is the size of the set of values of  $\gamma$  used in Eq. 12.

#### IV. INFERRING GRASPS FOR NOVEL OBJECTS

After acquiring the contact model and the hand configuration model the robot is now presented with a new query

object to grasp. The aim is that the robot finds a grasp such that its parts  $H_i$  are well-placed with respect to the object surface, while preserving similarity to the example grasps.

First of all we combine the contact models with the observed point cloud for the object to obtain a set of *query densities*, one for each hand part. The  $i^{\text{th}}$  query density  $Q_i$  is a density over where the  $i$ -th hand part  $H_i$  can be on the surface of new object (see Fig. 5). From the query densities, a hand pose is generated as follows. We randomly pick a hand part  $j$ . We randomly sample, from the corresponding query density, a pose for part  $j$ . We sample poses for all other hand parts, from the hand configuration model aligned to match the pose of part  $i$ , yielding a kinematically feasible hand configuration that is compatible with the pose selected for part  $i$ . We refine the grasp by performing a simulated annealing search in the hand configuration space, to locally maximise the grasp likelihood measured as the product of the hand configuration density and the query densities for all the hand parts, thereby ensuring adjustments of the remaining hand parts to make contacts with surface patches of similar local curvature to their contact models. We repeat the entire process a fixed number of times, and select the most likely kinematically feasible grasp.

The optimisation procedure generates several possible grasps, each with its likelihood. Each grasp has a set of hand part poses that independently comply with the contact models, while jointly complying to the hand configuration model. The following subsections explain in detail how to create approximate query densities for a given query object, and how grasp optimisation is carried out using simulated annealing.

##### A. Query Density

This section explains how query densities are constructed. A query density emerges from the combination of a contact model with an object point cloud  $O$ . The  $i^{\text{th}}$  query density  $Q_i$  models where the  $i$ -th hand part  $H_i$  can be placed to grasp the object, given the features observed on the object's surface. Specifically,  $Q_i$  models the pose distribution of  $H_i$  in the world frame.

The contact model  $M_i$  models viable poses for  $H_i$  relatively to surface features. Intuitively, a contact model is constructed by finding pairs of matches between the features that belong to  $M_i$  and the features of  $O$ , and representing the distribution of poses for  $H_i$  that are suggested by all matching pairs of features.

The above procedure can be described in probabilistic terms as an integration over  $(v, r) \in O$  and  $(u, r) \in M_i$ , yielding query density  $Q_i$ :

$$Q_i(s) \equiv Z_i \int P_i(s|v, u) O(v, r) M_i(u, r) dv du dr \quad (13)$$

where  $Z_i$  is a normalisation constant, while  $O$  and  $M_i$  satisfy

$$\int O(v, r) dv dr = 1, \quad \int M_i(u, r) du dr = 1 \quad (14)$$



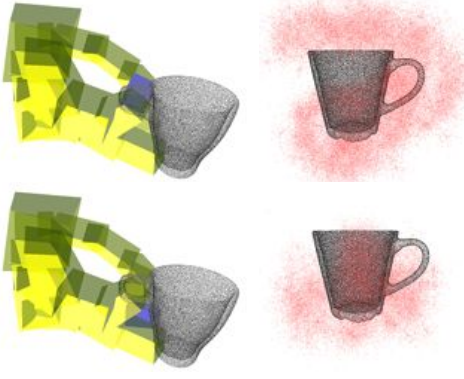


Fig. 5. Visualisation of two query densities (right) for two contact models (left) of a handle grasp. Each contact model is created for a single hand part (blue) for an example mug (left). A query density is a distribution over poses of the corresponding hand part (red cloud) for a new “query” mug.

Furthermore, the conditional density  $P_i$  represents pose multiplication  $s = v \circ u$ ,  $O$  and  $M_i$  contribute to the weight of pose  $s$  through the common feature variable  $r$ .

Using Bayes rule the integral (13) can be rewritten as follows:

$$Q_i(s) = Z_i \int P_i(s|v, u) O(r|v) O(v) M_i(u|r) M_i(r) dv du dr \quad (15)$$

If the involved objects are represented simply as a collection of surface features (5) without uncertainties,  $O(r|v)$  becomes a mapping  $r(v) : v \rightarrow r$  which for any choice of  $v$  uniquely determines  $r$ , effectively removing integration over  $r$ . Eq. (15) can be approximated as

$$Q_i(s) \simeq \int P_i(s|v, u) O(v) M_i(u|r(v)) M_i(r(v)) dv du \quad (16)$$

Eq. (16) can be computed directly via Monte Carlo integration [5], [6], which generates a number of weighted kernels, by repeating the following steps:

- 1) Sample  $(\hat{v}_\ell, \hat{r}_\ell) \sim O$ , i.e. just pick up random surface feature from the point cloud
- 2) Sample from conditional density  $(\hat{u}_{i\ell}) \sim M_i(u|\hat{r})$
- 3) Compute kernel centre  $\hat{s}_{i\ell} = \hat{v}_\ell \circ \hat{u}_{i\ell}$
- 4) Compute kernel weight  $w_{i\ell} = M_i(\hat{r}_\ell)$

Query density (13) can be finally approximated as

$$Q_i(s) \simeq \sum_{\ell} w_{i\ell} \mathcal{N}_3(p|\mu_{\hat{p}_{i\ell}}, \sigma_{p_i}) \Theta(q|\mu_{\hat{q}_{i\ell}}, \sigma_{q_i}) \quad (17)$$

with  $\ell$ -th kernel centre  $(\mu_{\hat{p}_{i\ell}}, \mu_{\hat{q}_{i\ell}}) = \hat{s}_{i\ell}$ , and where all weights were normalised  $\sum_{\ell} w_{i\ell} = 1$ .

The bandwidths  $\sigma_{p_i}$  and  $\sigma_{q_i}$  in (17) were hand tuned and kept fixed in all the experiments.

Fig. 5 visualises two example query densities created for two contact models of a handle grasp (see also Fig. 10).

### B. Grasp Optimisation

We now describe how to compute the most desirable grasp configuration for a new object. Let us denote by  $\mathbf{s} = (s_1, \dots, s_N)$  the configuration of the hand in terms of a set of hand part poses  $s_i \in SE(3)$ . Let us also denote

by  $\mathbf{h} = (h_w, h_j)$  the hand pose in term of a wrist pose  $h_w \in SE(3)$  and joint configuration  $h_j \in \mathbb{R}^D$ . Finally, let  $k^{\text{for}}(\cdot)$  denote the forward kinematic function of the hand, with

$$\mathbf{s} = k^{\text{for}}(\mathbf{h}), \quad s_i = k_i^{\text{for}}(\mathbf{h}) \quad (18)$$

We infer a desirable set of grasp parameters by maximising a product

$$h = \underset{(h_w, h_j)}{\operatorname{argmax}} C(h_j) \prod_i Q_i(k_i^{\text{for}}(h_w, h_j)). \quad (19)$$

where  $C(h_j)$  is the hand configuration model (12) and  $Q_i$  are query densities (17).

In other words, we search for the hand pose  $\mathbf{h}$  which maximises the pose probability of hand parts  $H_i$  given surface features observed on the object, while limiting the search to hand configurations that resemble those from training.

Product 19 is maximised using a simulated annealing procedure which is initialised as follows:

- 1) pick up random hand part  $i$
- 2) find the corresponding part pose  $s_i$  by sampling from  $Q_i$
- 3) pick up random hand configuration  $h_j$  by sampling from  $C(h_j)$
- 4) compute remaining parts poses  $s$  and hand pose  $\mathbf{h}$  using forward kinematics
- 5) evaluate Eq. 19, retain best scoring  $\mathbf{h}$ .

In the next iterations, the simulated annealing procedure skips steps 1) and 2) directly generating wrist pose  $h_w$  in some neighbourhood of the previous best scoring one.

The entire procedure is then repeated 1000 times, generating a large number of solutions (grasps) which are then clustered and ranked, yielding a variety of candidate hand poses (“grasp clusters”) which are then checked against arm reachability and workspace constraints. The remaining best scoring hand pose is then used to generate an approach trajectory. The computation of the query distribution has time complexity  $\Omega(|r||O||L|)$  where  $L$  is the number of kernels in the query density. The query density is calculated once for each new object for each hand part prior to grasp optimisation.

## V. EXPERIMENTAL METHOD

We tested our method with 18 real objects and two robot platforms. We performed three experiments. In these we use two types of point cloud models of objects. Type 1 (a full object model) is a precise high density point cloud created by sampling from a CAD model of the object. Type 2 (a partial object model) is a point cloud gathered from two views of the object on the fly, using a depth camera, and is thus more sparse, and more noisy. In Experiment 1 we trained and tested grasps on objects with full models. In Experiment 2 we trained on objects with full models, but transferred to objects with partial models. In Experiment 3 we trained and tested on objects with partial models.

The method adapts a known grasp type to a novel object, and so the training set was organised by grasp type. Six

| Grasps on        | #Succ. | #Fail | Success rate |
|------------------|--------|-------|--------------|
| Training objects | 14     | 2     | 88%          |
| Novel objects    | 31     | 5     | 86%          |

TABLE I

ALL EXPERIMENTS: GRASP SUCCESS RATES.

grasp types were used: rim, pinch, pinch with support, power, handle grasp with support, and a top grasp. A training object was placed in front of the robot, and a partial point cloud constructed from two views. In Experiments 1 and 2 pose estimation for the full object model was then performed with a model fitting procedure [13]. In Experiment 3 only the partial point cloud gathered on the fly was used for training. The training grasp was then taught on the localised object model (Experiment 1,2) or the partial point cloud (Experiment 3), and the contact model, and hand configuration model learned. The training grasp was then executed on the real object. This real execution was not used for the learning, but to test the residual failure rate due to localisation errors. After training, test objects were presented in sequence, a grasp type selected for each by hand. For each test object the grasp adaptation procedure ran 1000 times to generate 1000 candidate grasps, taking a total of 5 secs to compute. These grasps were ranked by the score  $h$  (Eq. 19) and working down the list each was tested for kinematic feasibility using collision checking and reachability analysis for the final hand pose. The highest scoring feasible grasp was then selected to be performed. Two robot platforms were used. The first was the DLR Justin mobile robot (Fig. 11), equipped with a 15-DoF DLR II hand with 4 fingers. The second robot platform (Fig. 6) had a 20-DoF DLR-HIT II hand with 5 fingers.

## VI. RESULTS

Success rates aggregated over all three experiments are shown in Table I, 88% of the grasps executed on training objects succeeded, while 86% succeeded on novel objects. Success here corresponds to the ability to lift up an object at least 20cm above the table. The training case failures are due to localisation error. Table II shows the training and test sets for Experiment 1. For Experiment 2 we trained two grasps: power and pinch on the bottle and dumpling box respectively, and transferred those successfully the moisturiser and anti-perspirant (pinch), and spray tin (power) (Fig. 12). In Experiment 3 we trained on the stapler (rim) and successfully tested on the bowl and the cutlery holder (Fig. 13). Pictures of results for Experiment 1 are shown in (Figs. 8-10).

## VII. CONCLUSIONS AND DISCUSSION

In this paper we have presented a method that generalises a single grasp (contacts and trajectories) of an object to several grasps of other objects with different global shapes. One essential element of our approach is learning a separate *contact model* for each finger phalange how of its pose relative to the surface is related to local surface feature. Another is that we learn a *hand configuration model* based

| Platform  | Training  | Test  |
|-----------|---|---|
| DLR-HIT 2 | glass (r),<br>bottle (p), tube (ps)<br>dumpling box (w) | mug2 (r), bottle (ps),<br>toothpaste (p,ps), cup (r)<br>herring tin (w,ps),<br>chocsticks box (w),<br>drops box (ps), tube (p), |
| Justin    | mug1 (t,h)<br>bottle (ps)                               | mug2 (t,h), glass (t)<br>cup (t), tube (ps)   |

TABLE II

EXPERIMENT 1 SETUP. MUG1 AND 2 WERE ALSO SWAPPED. PINCH=P, RIM=R, PINCH SUPPORT=PS, POWER=W, TOP=T, HANDLE=H.

on sampling poses near to those on the approach trajectory in the training example.

The empirical studies we have performed show that: i) our method can learn from one or two example grasps of a particular type (e.g. power, pinch, pinch with support); ii) the system creates grasps for objects we tested with globally different shapes from the training objects, but which share some local shapes; iii) for a new object many new grasps can be generated, ordered by likelihood, and these form grasp clusters, allowing the selection of grasps that satisfy workspace constraints; iv) successful new grasps can also be generated even where shape recovery is incomplete for the new object; v) this model works for a total of 31 transferred grasps of novel objects made with two different multi-fingered robot systems.

Future planned work includes automating selection between types of trained grasps. At the moment we only show that a trained grasp type can generalise to new objects, and not that we can automatically choose a grasp type.

## REFERENCES

- [1] C. Bard and J. Troccaz. Automatic preshaping for a dextrous hand from a simple description of objects. In *IEEE IROS*, 1990.
- [2] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters. Generalization of human grasping for multi-fingered robot hands. In *IROS*, pages 2043–2050. IEEE, 2012.
- [3] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, 2000.
- [4] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergstrom, D. Kragic, and A. Morales. Mind the gap – robotic grasping under incomplete observation. In *IEEE ICRA*, 2011.
- [5] R.E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.
- [6] A. Chiasso and S. Soatto. Monte carlo filtering on lie groups. In *Proc. of IEEE CDC*, volume 1, pages 304–309, December 2000.
- [7] J. Coelho, J. Piater, and R. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. In *Robotics and Autonomous Systems*, volume 37, pages 7–8, 2000.
- [8] R. Detry, C. Henrik Ek, M. Madry, and D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE ICRA*, 2013.
- [9] D. Fischinger and M. Vincze. Empty the basket – a shape based learning approach for grasping piles of unknown objects. In *IEEE IROS*, 2012.
- [10] R. A. Fisher. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A.*, 1953.
- [11] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE ICRA*, 2012.
- [12] U. Hillenbrand and M.A. Roa. Transferring functional grasps through contact warping and local replanning. In *IROS*, pages 2963–2970. IEEE, 2012.

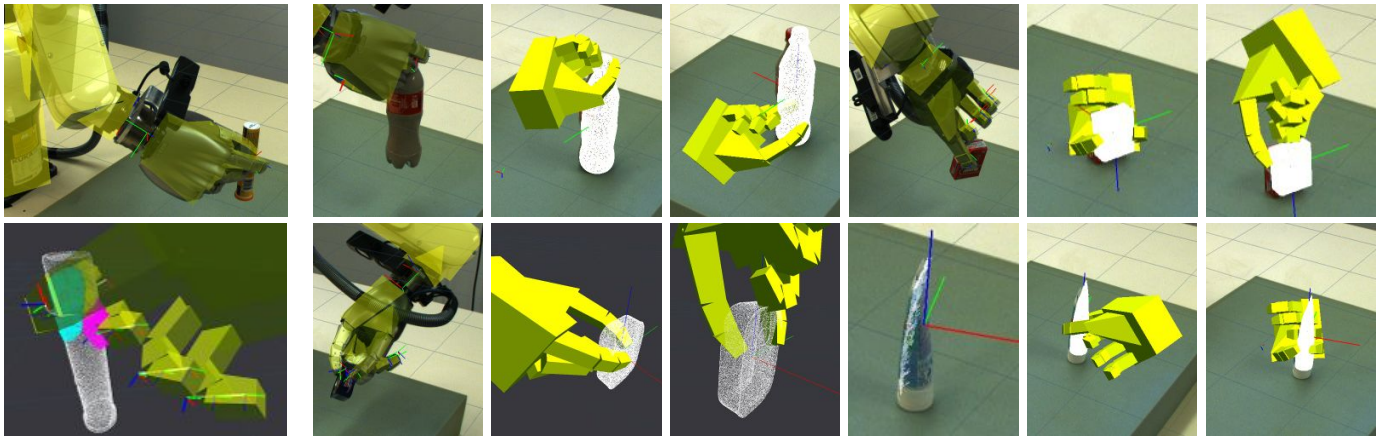


Fig. 6. Pinch with support grasp. The first column shows the demonstrated grasp and resulting learned contact models for a tablet tube. Remaining figures in middle and bottom rows show grasps subsequently generated by the agent for - coke bottle, orange-drops box and herring tin objects (executed grasps plus 2 alternative grasp clusters for each object), and a toothpaste tube object (executed grasp plus one alternative grasp cluster - both shown as simulator images due to poor camera viewing angle during execution).

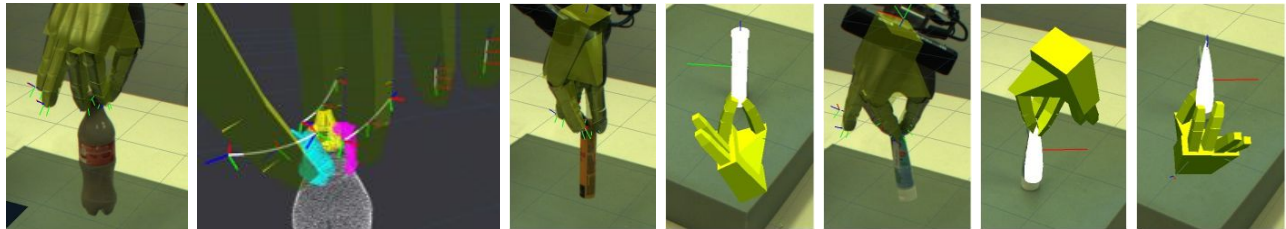


Fig. 7. Pinch grasp. Left to right: demonstrated grasp on coke bottle and resulting learned contact models; transferred grasp to tablet tube plus one additional grasp cluster; transferred grasp to toothpaste tube plus two additional grasp clusters.

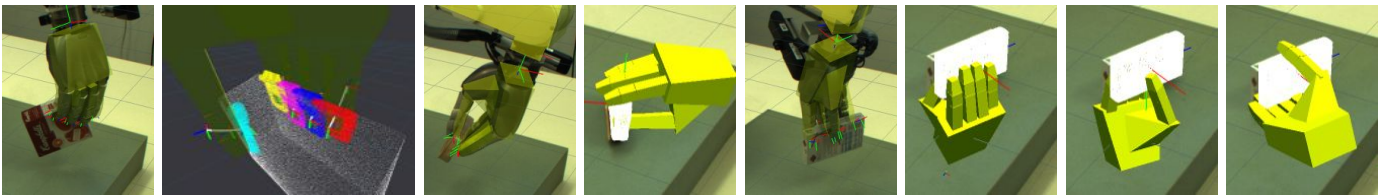


Fig. 8. Power grasp. Left to right: demonstrated grasp on instant dumplings box and learned contact models; transfer to herring tin (executed grasp plus additional grasp cluster); transfer to choosticks box (executed grasp plus three additional grasp clusters).

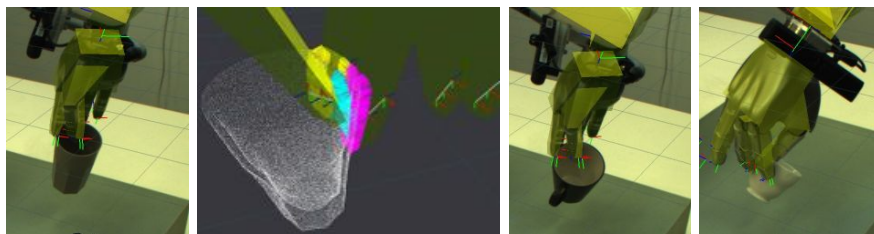


Fig. 9. Rim grasp. Left to right: demonstrated grasp on water glass and learned contact models; transferred grasps to two different tea-cups.

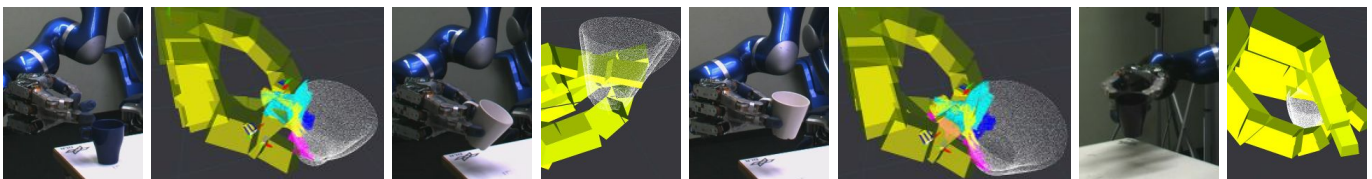


Fig. 10. Mug handle grasp. Left to right: demonstrated grasp mug 1 (fails!) and learned contact models; transfer to mug 2 (executed plus additional cluster); demonstrated grasp mug 2; transfer to mug 1 (executed plus additional cluster).



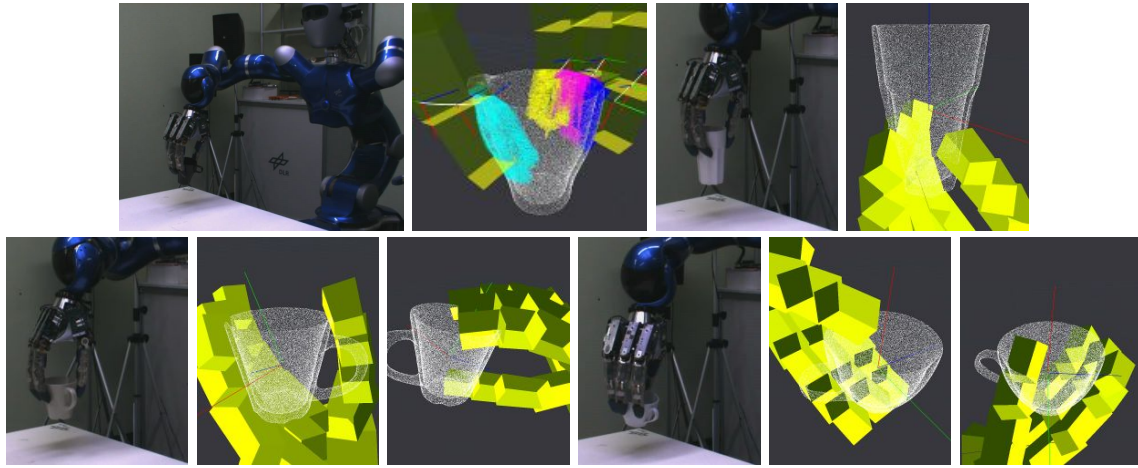


Fig. 11. Mug top grasp. Top row - demonstration and learned contact model for mug 1; transferred grasp plus one additional grasp cluster for water glass. Bottom row - transferred grasps plus two additional grasp clusters for mugs 2 and 3.

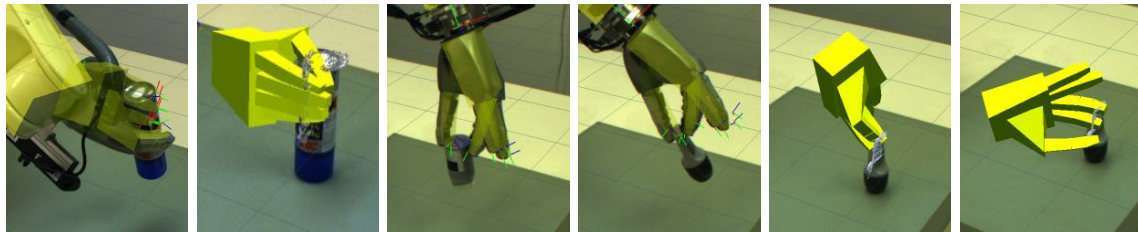


Fig. 12. Grasp transfers from complete point cloud models to objects represented by incomplete point clouds. Left to right: power grasp of lubricant spray (executed grasp plus one additional grasp cluster); pinch grasp on moisturiser; pinch grasp on antiperspirant (executed grasp plus two additional grasp clusters).

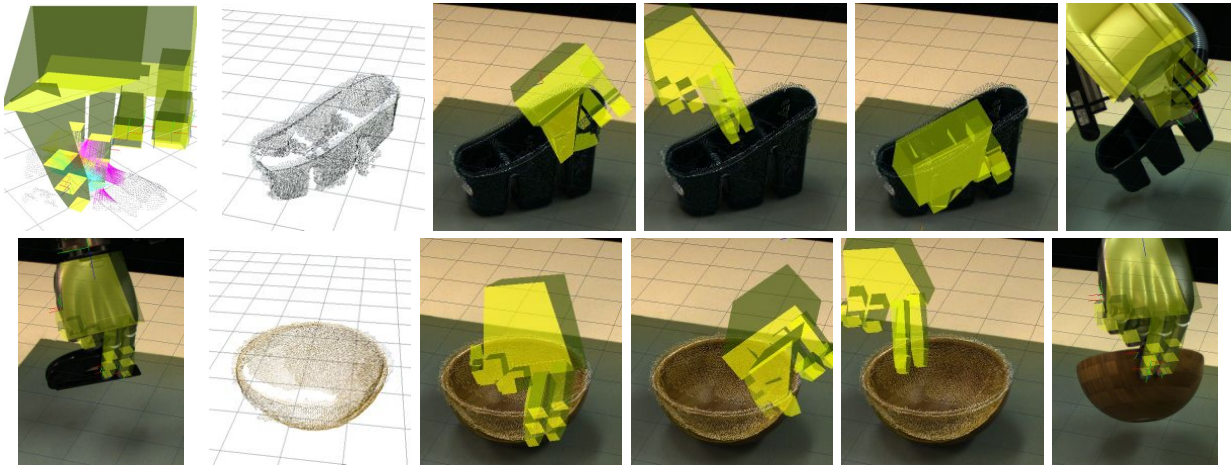


Fig. 13. Pinch grasp transfer from a stapler incomplete point cloud model (top and bottom left images) into two objects also represented by incomplete point clouds: a cutlery holder (top) and a wooden bowl (bottom).

- [13] Ulrich Hillenbrand. Non-parametric 3d shape warping. In *Pattern Recognition (ICPR)*, pages 2656–2659. IEEE, 2010.
- [14] O. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *RAS*, 58, 2010.
- [15] O. Kroemer, E. Ugur, E. Oztup, and J. Peters. A kernel-based approach to direct action perception. In *IEEE ICRA*, 2012.
- [16] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil. Using experience for assessing grasp reliability. *Int. Journ. of Humanoid Robotics*, 1(4):671–691, 2004.
- [17] M. Popović, D. Kraft, L. Bodenhagen, E. Başeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger. A strategy for grasping unknown objects based on co-planarity and colour information. *RAS*, 2010.
- [18] R. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA 2011*, Shanghai, China, May 9-13 2011.
- [19] A. Saxena, L. Wong, and A.Y. Ng. Learning grasp strategies with partial shape information. In *AAAI*, 2008.
- [20] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [21] M. Spivak. *A comprehensive introduction to differential geometry*, volume 1. Publish or Perish Berkeley, 1999.
- [22] Erik B. Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, MIT, Cambridge, MA, 2006.
- [23] S. Thrun and B. Wegbreit. Shape from symmetry. In *IEEE ICCV*, volume 2, pages 1824–1831, 2005.