The International Journal of Robotics Research
–(–):1–24

DOI:–
http://mms.sagepub.com

One shot learning and generation of dexterous grasps for novel objects

Marek Kopicki, Maxime Adjigble, Rustam Stolkin, Ales Leonardis
School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT.

Renaud Detry
University of Liège, Belgium

Jeremy L. Wyatt*
CN-CR, University of Birmingham, Edgbaston, Birmingham, B15 2TT and IRI, University Polytechnic de Catalunya, Barcelona

Abstract

This paper presents a method for one-shot learning of dexterous grasps, and grasp generation for novel objects. A model of each grasp type is learned from a single kinesthetic demonstration, and several types are taught. These models are used to select and generate grasps for unfamiliar objects. Both the learning and generation stages use an incomplete point cloud from a depth camera – no prior model of object shape is used. The learned model is a product of experts, in which experts are of two types. The first is a *contact model* and is a density over the pose of a single hand link relative to the local object surface. The second is the *hand configuration model* and is a density over the whole hand configuration. Grasp generation for an unfamiliar object optimises the product of these two model types, generating thousands of grasp candidates in under 30 seconds. The method is robust to incomplete data at both training and testing stages. When several grasp types are considered the method selects the highest likelihood grasp across all the types. In an experiment, the training set consisted of five different grasps, and the test set of forty-five previously unseen objects. The success rate of the first choice grasp is 84.4% or 77.7% if seven views or a single view of the test object are taken, respectively.

Keywords
learning, dexterous grasping

* Corresponding author; e-mail: jlw@cs.bham.ac.uk

Fig. 1. Leftmost image: Objects used, the four objects on the left were used solely for training, the remaining forty three objects on the right were solely used as novel test objects. **Rightmost image:** The Boris manipulation platform on which the experiments reported were carried out.

1. Introduction

Transferring dexterous grasps to novel objects is an open problem. In this paper we present a method that achieves this using as little as one training example per grasp type. The method enables learning and transfer of five grasp examples, including power and pinch-type grasps, to 43 unfamiliar test objects. Grasps generalise to test objects of quite different shape to the training object, for instance from a bowl to a kettle. The approach selects as well as adapts a grasp type from several learned types, simply selecting the type that enables the most similar grasp to training. The method copes with partial and noisy shape information for the test objects, and also generates different grasps for the same object presented in various orientations. The method requires no knowledge of the human defined object category either when learning or performing transfer.

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 (Saxena et al., 2008; Detry et al., 2013; Herzog et al., 2014; Kroemer et al., 2012). 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 (Ben Amor et al., 2012). This global hand shape can additionally be associated with global object shape, allowing generalisation by warping grasps to match warps of global object shape (Hillenbrand & Roa, 2012). This second class works well for high DoF hands, but 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 using a product of experts formulation when inferring a new grasp. Dexterous grasping involves simultaneously satisfying multiple constraints, and our central insight is that a product of experts is a natural way to encode these. Both model types are density functions. The first is a *contact model* of the relation between a rigid link of the hand, and the local object shape near its point of contact. We learn one contact model for each link of the hand involved in the grasp, and these capture local constraints in the grasp. To capture global information 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, grasps can be found for novel objects. When presented with a test object, a Monte Carlo procedure is used to combine a contact model with the available point cloud for the new object, to construct a third type of density function (called a *query density*). We build one query density for each hand link, and use them in two ways to find a grasp for the new object. 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 remaining link poses via forward kinematics. The whole solution is then refined using local search. The search seeks to maximise the product of experts involving each query density and the hand configuration density.

The paper is organised as follows. We begin with a survey of related work (Sec. 2). We then continue with a description of the representations employed and the learning process (Sec. 3), followed by a description of the process for finding a grasp for a novel object (Sec. 4). We finish with an experimental study (Sec. 5) and a discussion (Sec. 6).

2. Related work

In robotics, grasp planning is driven by two dominant trends. Traditional grasp planning relies on force analysis (FA), which computes the behaviour of an object subject to a grip via the laws of classical mechanics (Bicchi & Kumar, 2000). In recent years, a second trend emerged, whereby a direct mapping from vision to action is either engineered or learned from experience (Bard & Troccaz, 1990; Coelho et al., 2000; Kamon et al., 1996). When comparing one approach to the other, force analysis is the methodical, scrupulous approach, where one attempts to model the physical processes that occur at the interface of the object and the gripper. Given a model of the shape, weight distribution, and surface friction of an object, a model of the shape, kinematics, and applicable forces/torques of a gripper, and a model of object-gripper contacts, force analysis applies the laws of classical mechanics to compute the magnitude of the external disturbances that a grasp can withhold. In turn, from the range of disturbances that a grasp can withhold, authors have defined a number of so-called grasp quality metrics (Shimoga, 1996), amongst which stands the famous *epsilon* measure of Ferrari & Canny (1992). A grasp is called force-closure if external forces can be balanced by the gripper, thereby restraining the object within the hand.

To plan a grasp via force analysis, several problems must be solved. The robot first needs to build a representation of the object. If the object is seen from a single viewpoint, the robot needs either to access to a representation of the complete object shape, or to hypothesise the shape of the occluded side of the object. The robot must also make a fair assessment of the object's mass, mass distribution and friction coefficient (Zheng & Qian, 2005; Shapiro et al., 2004). Then, it attempts to find a set of points on the object's surface that are such that if the gripper's fingers were contacting the object at those points, the grasp would be stable (Shimoga, 1996; Pollard, 2004; Liu, 2000). To compute the forces that are applicable at those points, the robot must rely on a model of hand-object contacts, in part to assess the amplitude of friction forces. The deformation of the object's surface around a contact point is hard to predict. One often assumes hard contacts with a fixed contact area (Bicchi & Kumar, 2000). One also usually assumes static friction between the hand and the object (Shimoga, 1996). Finally, the robot verifies that the grasp is kinematically feasible, i.e., that the hand can be moved to a configuration that realises those contacts (Rosales et al., 2011). Force analysis is applicable to multi-fingered hands, and its ability to generate complex grasps has been shown in the literature (Boutselis et al., 2014; Gori et al., 2014; Grupen, 1991; Hang et al., 2014; Rosales et al., 2012; Saut & Sidobre, 2012; Xu et al., 2007).

Despite its strong theoretical foundation and conceptual elegance, force analysis has not solved robot grasping entirely. FA is difficult to use in an open-ended environment where perceiving and acting are subject to high degrees of noise. Small errors in estimating the pose of an object or in moving the gripper to its intended position can lead to a grasp whose quality substantially differs from the intended one (Zheng & Qian, 2005). This problem can be mitigated by computing *independent contact regions* (ICRs) (Ponce & Faverjon, 1995), i.e., maximal segments of the object's surface where the fingers can be applied while maintaining force closure. Yet, ICRs still suffer from other shortcomings of force analysis, such as difficulties in estimating shape, mass or friction parameters (Rusu et al., 2009).

Research has shown on several occasions that the correlation between grasp quality metrics and real-world grasp outcomes is limited (Bekiroglu et al., 2011; Kim et al., 2013; Goins et al., 2014). In addition to these limitations, FA remains a computationally-expensive method. These considerations have encouraged researchers to explore different means of planning grasps. As mentioned above, many have begun studying means of building a direct mapping from vision to action, closer in spirit to the way primates establish grasping plans (Jakobson & Goodale, 1991; Hu et al., 1999; Rizzolatti & Luppino, 2001; Fagg & Arbib, 1998; Borra et al., 2011). The mapping captured implicitly by a learning method has merit of its own. It can in some situations be complementary to force analysis, and in other situations be entirely sufficient to perform robot grasping.

Within the class of methods that do not rely on force analysis, a first group plans grasps by searching for shapes that fit within the robot's gripper (Fischinger & Vincze, 2012; Popović et al., 2010; Trobina & Leonardis, 1995; Klingbeil et al., 2011; Richtsfeld & Zillich, 2008; Kootstra et al., 2012; ten Pas & Platt, 2014). Popović et al. (2010) computed grasps onto object edges detected in 2D images, by defining rules such as "*two parallel edges can be grasped by placing two fingers on the outer sides of both edges*". Klingbeil et al. (2011) searched through range data for sites where the PR2's two-finger gripper fits an object, by considering planar sections of the 3D image and identifying U-shaped boundaries that resemble the inside of the PR2 gripper. Such methods work well with simple grippers, but with more complex grippers, the number of rules that need to be hard-coded for the gripper to work well with objects of different sizes and shapes quickly becomes unmanageable. This problem can be overcome by letting the robot learn the mapping from vision to action (Coelho et al., 2000; Kamon et al., 1996; Morales et al., 2004; Platt et al., 2006; Bard & Troccaz, 1990; Detry et al., 2013; Herzog et al., 2014; Kroemer et al., 2012, 2010; Saxena et al., 2008; Zhang et al., 2011; Kim, 2007), instead of hard-coding it. The vision domain of the mapping has been parametrised by features such as SIFT (Saxena et al., 2008), 3D shape primitives (Platt et al., 2006), or 3D object parts (Kroemer et al., 2012; Detry et al., 2013). The action side of the mapping has been parametrised with a 3D grasping point (Saxena et al., 2008), a 6D gripper pose (Herzog et al., 2014) possibly accompanied by a hand pre-shape (Detry et al., 2013), or gripper-object contact points (Ben Amor et al., 2012). In our work, the robot learns a mapping from simple local 3D shape features to a complete parametrisation of the robot hand pose and its fingers.

Grasp learning algorithms can also classified according to the type of input they require. One class of methods focuses on learning a mapping from an image taken from a single viewpoint, to grasp parameters (Bard & Troccaz, 1990; Detry et al., 2013; Herzog et al., 2014; Kroemer et al., 2012, 2010; Saxena et al., 2008; Zhang et al., 2011; Kim, 2007). The image is provided by a depth sensor such as the Kinect, or by a stereo camera, and by nature it covers only one side of the object. Another class assumes the existence of a full 3D model of the object's shape (Hillenbrand & Roa, 2012; Ben Amor et al., 2012). Assuming a complete object model facilitates the planning problem, but it makes perception more challenging, as the robot is required to circle around a novel object before grasping it, and in many cases even then a complete model will not be obtained. Our method is designed to work with a setup that resides between these two classes: grasps are computed from an image captured from a single standpoint, by fixing the camera to the robot's arm and merging several views acquired from various extensions of the arm. We present experiments where the robot uses one to seven images captured from viewpoints spanning up to approximately 200° around the object.

Methods close in spirit to our own include the work of Hillenbrand & Roa (2012), who 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. (2012) 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. 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 nearly complete shape model of the target object is obtainable by sensing, but it is also applicable to partially-modelled objects, based one one view recovering as little as 20% of the object surface in our experiments. One difference in performance compared to the approach of Hillenbrand & Roa (2012) is that they transfer grasps within the same human defined object shape category (e.g. from one mug to another), whereas we are able to transfer grasps to different human defined object categories. Saxena et al. (2008) learned a three-finger grasp success classifier from a bank of photometric and geometric object features including symmetry, centre of mass and local planarity. Kroemer et al. (2012) and Detry et al. (2013) let the robot learn the pose and pre-shape of the hand with respect to object parts, and relied on compliance or force sensing to close the hand. Alternatively, Kroemer et al. (2010) also relied on control policies to adapt the fingers to the visual input. In our work, the robot plans a set of configurations for each finger individually, using local surface data, then it searches within those configurations for one that complies to hand kinematics. The result is an ability to plan dexterous multi-fingered grasps, while allowing for generalization of grasp models to objects of novel shape. In our earlier work Kopicki et al. (2014) we showed only how to solve the problem of adapting a particular grasp type, given a prior point cloud model of the object. Thus this current work goes beyond our previous work in that we now also: i) present a method to select between adapted grasp types, enabling automatic grasping of a much wider range of objects, and of objects presented in many orientations; ii) present extensive results for learning and testing without prior point clouds; iii) present testing with a variety of numbers of views of the test object.

3. Representations

This section describes the representations underpinning our approach. First we describe the kernel density representation that underpins all the models. The representation of the surface features necessary to encode the contact models follows. Finally we describe the form of the contact model and the hand configuration model. In the rest of the paper we assume that the robot's hand is formed of N_L rigid *links*: a palm, and a number of finger phalanges or links. We denote the set of links $L = \{L_i\}$. The representations are summarised at a high level in a video attached as Extension 2.

3.1. Kernel Density Estimation

Much of our work relies on the probabilistic modelling of surface *features*, extracted from 3D object scans. Features are composed of a 3D position, a 3D orientation, and a 2D local surface descriptor. Let us denote by $SO(3)$ 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* (a 3D position and 3D orientation), and surface descriptors are composed of two real numbers.

This paper makes extensive use of probability density functions (PDFs) defined on $SE(3) \times \mathbb{R}^2$. This section explains how we define these density functions. We represent PDFs non-parametrically with a set of K features (or particles) x_j

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

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* (Silverman, 1986), by assigning a kernel function $\mathcal{K}$ to each particle supporting the density, as

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

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 on the three components of our domain, namely $\mathbb{R}^3$, $SO(3)$, and $\mathbb{R}^2$. Let us denote the separation of feature x into $p \in \mathbb{R}^3$ for position, a quaternion $q \in SO(3)$ for orientation, $r \in \mathbb{R}^2$ for the surface descriptor. Furthermore, let us denote by μ another feature, and its separation into position, orientation and surface descriptor. Finally, we denote by σ a triplet of real numbers:

$$x = (p, q, r), \tag{3a}$$

$$\mu = (\mu_p, \mu_q, \mu_r), \tag{3b}$$

$$\sigma = (\sigma_p, \sigma_q, \sigma_r). \tag{3c}$$

We define our kernel as

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

where μ is the kernel mean point, σ is the kernel bandwidth, and where $\mathcal{N}_n$ 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 $SO(3)$ (for details see (Fisher, 1953; Sudderth, 2006)). 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.

We note that thanks to the nonparametric representation used above, conditional and marginal probabilities can easily be computed from Eq. (2). The marginal density $\mathbf{pdf}(r)$ is computed as

$$\mathbf{pdf}(r) = \iint \sum_{j=1}^{K} w_j \mathcal{N}_3(p|p_i, \sigma_p)\Theta(q|q_i, \sigma_q)\mathcal{N}_2(r|r_i, \sigma_r)\mathrm{d}p\mathrm{d}q = \sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r), \tag{6}$$

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

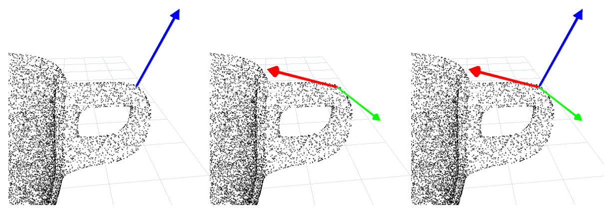
$$\mathbf{pdf}(p, q|r) = \frac{\mathbf{pdf}(p, q, r)}{\mathbf{pdf}(r)} = \frac{\sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r)\mathcal{N}_3(p|p_j, \sigma_p)\Theta(q|q_j, \sigma_q)}{\sum_{j=1}^{K} w_j \mathcal{N}_2(r|r_j, \sigma_r)}. \tag{7}$$

3.2. Surface Features

This section explains how the surface features discussed above are acquired from real object data. All objects considered in the paper are represented by point clouds constructed from one or multiple shots taken by a depth camera. A depth camera captures a set of points distributed in a 3D space along the object's visible surface. We directly augment these points with a surface normal and a curvature descriptor. As a result, the point clouds discussed below are composed of points that belong to $SE(3) \times \mathbb{R}^2$. As in the previous section, we denote a point of $SE(3) \times \mathbb{R}^2$ by x, and its separation into position-orientation-curvature components as p, q, and r. For compactness, we also denote the pose of a feature (its position and orientation) as v. As a result, we have

$$x = (v, r), \qquad v = (p, q). \tag{8}$$

The surface normal at p is computed from the nearest neighbours of p using a PCA-based method (e.g. (Kanatani, 2005)). Surface descriptors corresponds to the local *principal curvatures* (Spivak, 1999). The curvature at point p is encoded 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, $k_1 \in \mathbb{R}^3$, is a direction of the highest curvature. The second direction, $k_2 \in \mathbb{R}^3$, is perpendicular to k_1. The curvatures along k_1 and k_2 are denoted by $r_1 \in \mathbb{R}$ and $r_2 \in \mathbb{R}$ respectively, forming a 2-dimensional feature vector $r = (r_1, r_2) \in \mathbb{R}^2$. The surface normals and principal directions allow us to define the 3D orientation q that is associated to a point p. Fig. 2 illustrates a point's surface normal and curvature.



(a) (b) (c)

Fig. 2. An example object point cloud (black dots) with a selected point, its surface normal (blue axis), direction of the first principal curvature k_1 (red axis), and direction of the second principal curvature k_2 (green axis). These form a frame of reference depicted in the rightmost image.

The procedure described above allows the computation of a set of K_O features $\{(v_j, r_j)\}$ from a given object point cloud. In turn, the set of features defines a joint probability distribution, further referred to as the *object model*:

$$O(v,r) \equiv \mathbf{pdf}^O(v,r) \simeq \sum_{j=1}^{K_O} w_j \mathcal{K}(v,r|x_j,\sigma_x) \tag{9}$$

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 $w_j = 1/K_O$.

We note that the values computed for surface normals and curvatures are subject to ambiguities. For instance, there are always two ways of defining the directions of vectors k_1 and k_2 given a surface normal: (k_1, k_2) and $(-k_1, -k_2)$. For a sphere or a plane there are an infinite number of orientations about the normal. Finally for a point lying on a near-flat surface, the orientations of k_1 and k_2 within the tangent plane are also uncertain because of sensor noise. We account for these ambiguities/uncertainties at the stage of point cloud processing, by randomly sampling a direction or orientation amongst solutions. In this way, the ambiguity/uncertainty of normals and curvatures is represented by the statistics of the surface features that become the input data to the object model density. We now describe how we model the relationship of a finger link to the surface of the training object.

3.3. Contact Model

A contact model M_i encodes the joint probability distribution of surface features and of the 3D pose of the i-th hand link. Let us consider the hand grasping some given object. The (object) contact model of link L_i is denoted by

$$M_i(U,R) \equiv \mathbf{pdf}_i^M(U,R) \tag{10}$$

where M_i is short for $\mathbf{pdf}_i^M$, R is the random variable modelling surface features, and U models the pose of L_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 L_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 $\{x_j\}_{j=1}^{K_O}$, with $x_j = (v_j, r_j)$ and $v_j = (p_j, q_j)$, a contact model M_i is constructed from features from the object's surface. Surface 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 squared distance to the i^{th} link (Fig. 4). Additionally, features that are further than a cut-off distance δ_i from L_i are ignored. The weight is given by

$$w_{ij} = \begin{cases} \exp(-\lambda||p_j - a_{ij}||^2) & \text{if } ||p_j - a_{ij}|| < \delta_i \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

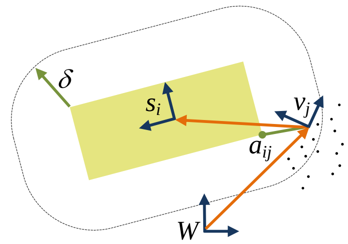
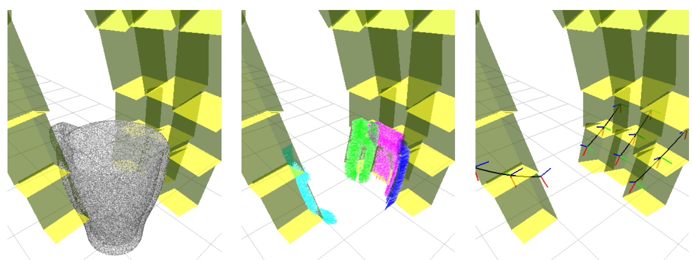


Fig. 3. Contact model. The figure shows the i-th link L_i (in yellow) and its pose s_i. The black dots are features from the surface of an object. The distance a_{ij} between feature v_j and the closest point on the link's surface is shown in green. The rounded rectangle illustrates the cut-off distance δ_i. The poses v_j and s_i are expressed in the world frame W. The top orange arrow illustrates u_{ij}, i.e., the pose of L_i relative to v_j.



(a) (b) (c)

Fig. 4. Example top grasp of a mug represented by a point cloud (a). The dotted coloured regions are rays between features and the closest hand link surfaces (b). The black curves with frames at the fingertips represent the range of hand configurations in Eq. (16) (c).

where $\lambda \in \mathbb{R}^+$ and a_{ij} is the point on the surface of L_i that is closest to p_j.

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

$$u_{ij} = v_j^{-1} \circ s_i, \tag{12}$$

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. 3). The contact model is estimated as

$$M_i(u, r) \simeq \frac{1}{Z} \sum_{j=1}^{K_{M_i}} w_{ij} \mathcal{N}_3(p|p_{ij}, \sigma_p) \Theta(q|q_{ij}, \sigma_q) \mathcal{N}_2(r|r_j, \sigma_r) \tag{13}$$

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 contact models are quite different for the different links within a grasp. This can be seen by comparing the marginalised contact models $M(r)$ for two example training grasps and two links in Fig. 5.

The parameters λ and σ_p, σ_q, σ_r were chosen empirically and kept fixed in all experiments reported in Sec. 5. 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.

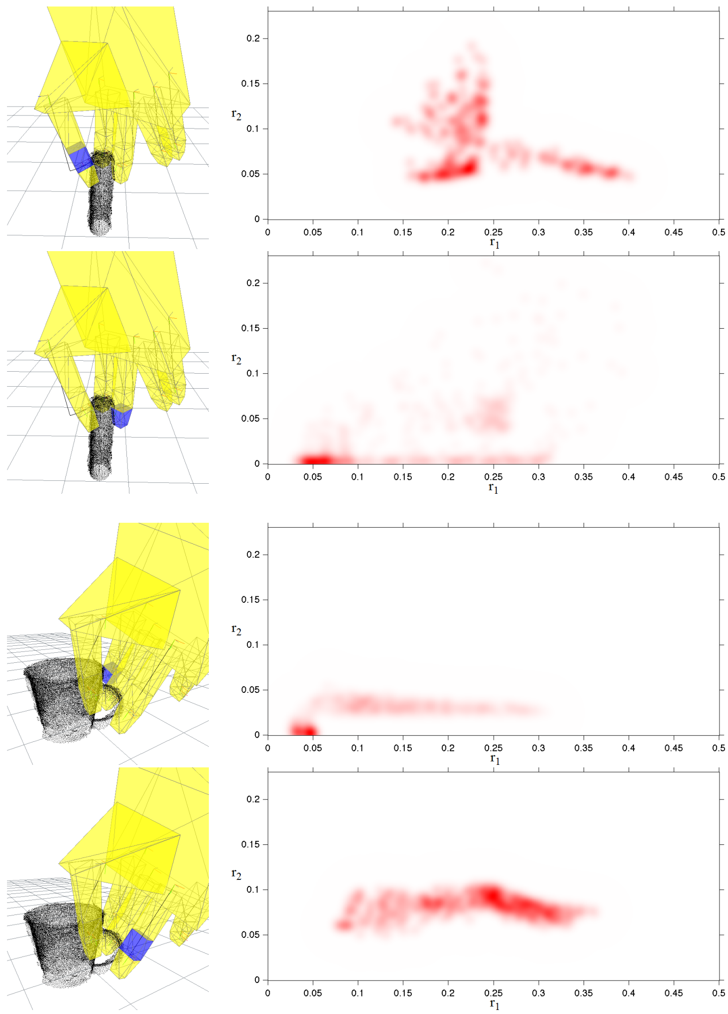


Fig. 5. Illustration of the contact model. Each image from the left column shows an example training grasp and a single selected link in blue colour. The corresponding image from the right column shows a distribution of the curvature descriptor for the features involved in the contact model of the selected link. The top two rows show contact models for two links for a pinch grasp on a vitamin tube, while the bottom two rows show contact models for two links from a handle grasp.

3.4. Hand Configuration Model

The hand configuration model, denoted by C, encodes a set of configurations of the hand joints $h_c \in \mathbb{R}^D$ (i.e., joint angles), that are particular to a grasp example. 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_c^t the joint angles at some small distance *before* the hand reached the training object, and by h_c^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_c^t and h_c^g, and extrapolated beyond h_c^g, as

$$h_c(\gamma) = (1 - \gamma)h_c^g + \gamma h_c^t \tag{14}$$

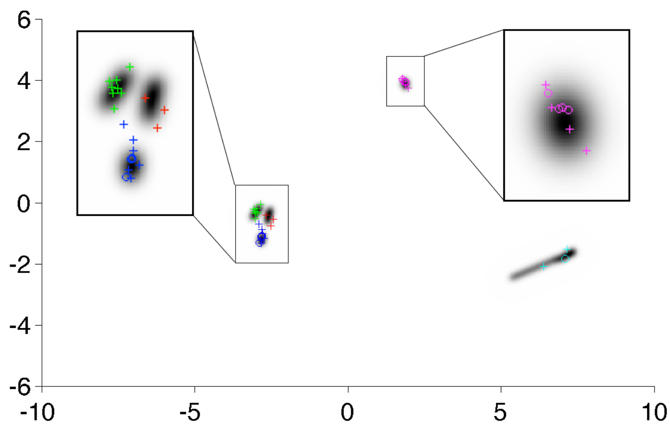


Fig. 6. Configuration models learned from real data (see Sec. 5). A configuration model is a PDF defined on the space of hand joint angles. The 20 degrees of freedom of our robot's hand make it difficult to plot a configuration model. Instead, we applied weighted PCA to the hand data collected during five different training grasps, and plot a KDE of the two principal components. The resulting density is shown in black in the figure. The coloured crosses show the configuration of grasps on test objects computed in Sec. 5 from the learned grasp models. Crosses coloured in red, green, blue, magenta and cyan respectively correspond to transferred grasps of type "handle", "pinch", "pinch with support", "power", and "powertube".

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

$$\mathcal{H}_c = \{h_c(\gamma) : \gamma \in [-\beta, \beta], \beta \in \mathbb{R}^+\}, \tag{15}$$

as

$$C(h_c) \equiv \sum_{\gamma \in [-\beta,\beta]} w(h_c(\gamma)) \mathcal{N}_D(h_c | h_c(\gamma), \sigma_{h_c}) \tag{16}$$

where $w(h_c(\gamma)) = \exp(-\alpha \| h_c(\gamma) - h_c^g \|^2)$ and $\alpha \in \mathbb{R}^+$. α and β were hand tuned and kept fixed in all the experiments. The hand configuration model computation has time complexity $\Omega(d_h K_C)$ where d_h is the number of dimensions of the configuration vector, and K_C is the size of the set of values of γ used in Eq. (16). Fig. 6 shows a plot of the configuration models learned in our experiments.

4. Inferring Grasps for Novel Objects

After acquiring the contact model and the configuration model, the robot is now presented with a new query object to grasp. The aim is that the robot finds a generalisation of a training grasp such that its links are well-placed with respect to the object surface, while preserving similarity to the example grasp. We infer generalised grasps for every example grasp, and pick the transfer grasp that is most likely according to the learned models.

First of all we combine each of the contact models with the query object's perceived point cloud, to obtain a set of *query densities*, one for each link that has an associated contact model. The i-th query density Q_i is a density modelling where the i-th link can be placed, with respect to the surface of a new object (see Fig. 7). From the query densities, a hand pose is generated as follows. We randomly pick a link i. We randomly sample, from the corresponding query density Q_i, a pose for link i. We sample, from the configuration model C, a hand configuration that is compatible with the pose selected for link i, and then we compute from forward kinematics the 3D poses of all the remaining hand links. We refine the grasp by performing a simulated annealing search in the hand configuration space, to locally maximise the grasp likelihood

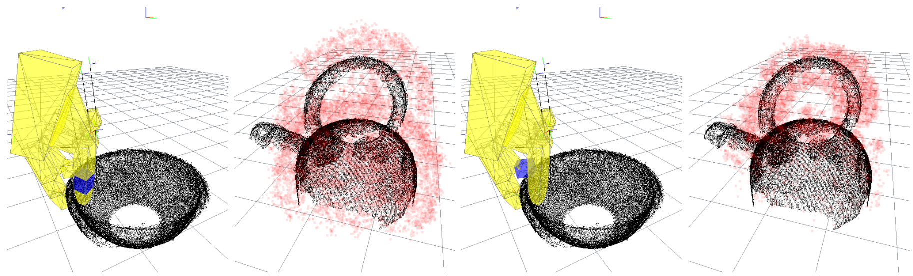


Fig. 7. Visualisation of two query densities (panels 2 and 4 from the left) for two contact models (panels 1 and 3) of a pinch with support grasp. The links to which the contact models are associated are in blue. A query density is a distribution over poses of the corresponding link (red cloud) for a new "query" kettle.

measured as the product of the hand configuration density and the query densities for all the hand links. We repeat the entire process a number of times, and select the most likely grasp that is also kinematically feasible.

The optimisation procedure generates many possible grasps, each with its likelihood. Each grasp has a set of link poses that independently comply with the contact models, while jointly complying with the hand configuration model. The following subsections explain in detail how to estimate query densities for a given query object, and how grasp optimisation is carried out. Extension 2 contains a high level video description of the grasp inference process as described in detail here.

4.1. Query Density

This section explains how query densities are constructed. A query density results from the combination of a contact model for a specific finger link with an object point cloud O for the new object. The purpose of a query density is both to generate and evaluate poses of the corresponding finger link on the new object. The i-th query density Q_i models the pose s *in the world frame* of the i-th link L_i.

A query density should be defined in a way that achieves good generalisation from training to test objects. To achieve this there are three relevant random variables to consider: the random variable V denoting a point on the object's surface, expressed in the world frame; the random variable for surface curvature of such a point R; and the random variable denoting finger link pose U relative to a local frame on the object. We define a joint density over all four variables, $\mathbf{pdf}_i(s, u, v, r)$. The pose distribution of robot link L_i is then defined by marginalisation with respect to u, v and r:

$$\mathbf{pdf}_i(s) = \iiint \mathbf{pdf}_i(s, u, v, r)\mathrm{d}v\mathrm{d}u\mathrm{d}r \tag{17}$$

Since $s = v \circ u$ it is completely determined by v and u. Thus we may factorise Eq. (17) as follows:

$$\mathbf{pdf}_i(s) = \iiint \mathbf{pdf}(s|u, v)\mathbf{pdf}_i(u, v, r)\mathrm{d}v\mathrm{d}u\mathrm{d}r \tag{18}$$

where $\mathbf{pdf}(s|u, v)$ is a Dirac delta function. We can factor Eq. (18) again by assuming that v (the density in the world frame of a surface point) and u (the distribution of the finger link pose relative to its closest surface point) are conditionally independent given r (the local curvature for a surface point):

$$\mathbf{pdf}_i(s) = \iiint \mathbf{pdf}(s|u, v)\mathbf{pdf}_i(u|r)\mathbf{pdf}(v|r)\mathbf{pdf}(r)\mathrm{d}v\mathrm{d}u\mathrm{d}r \tag{19}$$

where $\mathbf{pdf}_i(u|r) = M_i(u|r)$ and $\mathbf{pdf}(v|r) = O(v|r)$, since the object model does not depend on link L_i. The remaining question is how to determine the density over the curvatures r. Clearly to encourage generalisation curvatures should be preferred that are present in both the contact model and the object model, thus we set $\mathbf{pdf}(r) = M_i(r)O(r)$. Thus the i-th query density Q_i can be approximated by a single integral:

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 $\{(\hat{p}_{ij}, \hat{q}_{ij}, w_{ij})\}, \forall j$

$$Q_i(s) = \mathbf{pdf}_i(s) = \iiint T(s|u, v)M_i(u|r)O(v|r)M_i(r)O(r)\mathrm{d}v\mathrm{d}u\mathrm{d}r \tag{20a}$$

$$= \iiint T(s|u, v)O(v, r)M_i(u|r)M_i(r)\mathrm{d}v\mathrm{d}u\mathrm{d}r \tag{20b}$$

where $T(s|u, v) \equiv \mathbf{pdf}(s|u, v)$ which is the Dirac delta function mentioned above. Eq. (20) defines the density that must be computed for each link prior to grasp optimisation (Sec. 4.2). This query density (20) can be approximated by K_{Q_i} kernels centred on the set of weighted, sampled finger link poses for link L_i returned by Algorithm 1:

$$Q_i(s) \simeq \sum_{j=1}^{K_{Q_i}} w_{ij} \mathcal{N}_3(p|\hat{p}_{ij}, \sigma_{p_i})\Theta(q|\hat{q}_{ij}, \sigma_{q_i}) \tag{21}$$

with j-th kernel centre $(\hat{p}_{ij}, \hat{q}_{ij}) = \hat{s}_{ij}$, and where all weights were normalised $\sum_j w_{ij} = 1$. The number of kernels $K_{Q_i} = K_Q$ were chosen equal for all query densities and grasp types (unless otherwise stated), also the bandwidths σ_{p_i} and σ_{q_i} in Eq. (21) were hand tuned and kept fixed in all the experiments. Fig. 7 depicts two example query densities created for two contact models of a handle grasp.

When a test object is presented a set of query densities $\mathcal{Q}^g$ is calculated for each training grasp g. The set $\mathcal{Q}^g = \{Q_i^g\}$ has $N_Q^g = N_M^g$ members, one for each contact model M_i^g in $\mathcal{M}^g$. The computation of each query density has time complexity $\Omega(K_{M_i} K_Q)$ where K_{M_i} is the number of kernels of the i-th contact model density (13), and K_Q is the number of kernels of the corresponding query density.

4.2. Grasp Optimisation and Selection

During testing the robot will have at its disposal N_G grasp types $\mathcal{G} = \{\mathcal{Q}^g, C^g\}$. We now describe how these are used to generate a set of ranked grasps for a new object by Algorithm 2. There is an initial grasp generation phase. This is followed by interleaved grasp optimisation and selection steps.

Grasp Generation A initial set of grasps is generated for each grasp type g by randomly picking a query density Q_i^g and then sampling a pose s_i from it. Together with a sample h_c from C^g this defines a complete hand pose h. A set of initial solutions across all grasp types $\mathcal{H}^1 = \{h_j^g\}$ is generated, where h_j^g means the j^{th} initial solution for grasp type g. To represent each of these grasp solutions let us denote by $s_{1:N_L} = (s_1, ..., s_{N_L})$ the configuration of the hand in terms of a set of hand link poses $s_l \in SE(3)$. Let us also denote by $h = (h_w, h_c)$ the hand pose in terms of a wrist pose $h_w \in SE(3)$

Algorithm 2: Grasp Optimisation and Selection ($\{\mathcal{Q}^g, C^g\}, \forall g, \mathcal{K}_{selection}$)

For each grasp g
  For $j = 1$ to N
    Randomly select a query density $Q_i^g$from $\mathcal{Q}^g$
    Sample the pose s_i of the i^{th} link from Q_i^g
    Sample a hand configuration h_c from $C^g(h_c)$
    Compute the remaining hand link poses and thus overall hand configuration and pose h_j^g using forward kinematics
  end
end
$\mathcal{H}^1 = \{h_1^1, \ldots, h_j^1, \ldots, h_N^1, h_1^2, \ldots, h_N^2, h_1^3, \ldots, \ldots, h_N^{N_g}\}$
For $k = 1$ to K
  if $k \in \mathcal{K}_{selection}$
    rank $\mathcal{H}^k$ by Eq. 26 and retain top $p\%$
  end
  for $m = 1$ to $|\mathcal{H}^k|$
    $\mathcal{H}_m^k$ = perform a step of simulated annealing on $\mathcal{H}_m^k$ using Eq. 23 as the objective function.
  end
  $\mathcal{H}^{k+1} = \mathcal{H}^k$
end
rank $\mathcal{H}^{K+1}$ by Eq. 26
return $\mathcal{H}^{K+1}$

and joint configuration $h_c \in \mathbb{R}^D$. Finally, let $k^{\text{for}}(\cdot)$ denote the forward kinematic function of the hand, with

$$s_{1:N_L} = k^{\textbf{for}}(h), \quad s_l = k_l^{\text{for}}(h) \tag{22}$$

Having generated an initial solution set $\mathcal{H}^1$ stages of optimisation and selection are interleaved.

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

$$\underset{(h)}{\text{argmax}}\, \mathcal{L}^g(h) = \underset{(h)}{\text{argmax}}\, \mathcal{L}_C^g(h)\mathcal{L}_Q^g(h) = \underset{(h_w, h_c)}{\text{argmax}}\, C^g(h_c) \prod_{Q_i^g \in \mathcal{Q}^g} Q_i^g\left(k_i^{\text{for}}\left(h_w, h_c\right)\right) \tag{23}$$

where $\mathcal{L}^g(h)$ is the overall likelihood, where $C^g(h_c)$ is the hand configuration model (16), Q_i^g are query densities (21). Improvement is by simulated annealing (SA) (Kirkpatrick et al., 1983). 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 $\mathcal{H}^k$.

Grasp Selection Steps During periodic, predetermined selection steps, grasps are ranked and only the most likely $p\%$ retained for further optimisation. During these selection steps the criterion in (23) is augmented with an additional expert $W(h_w, h_c)$ penalising collisions in a soft manner. This penalises grasps which are likely to lead to grasp failure. 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. 23:

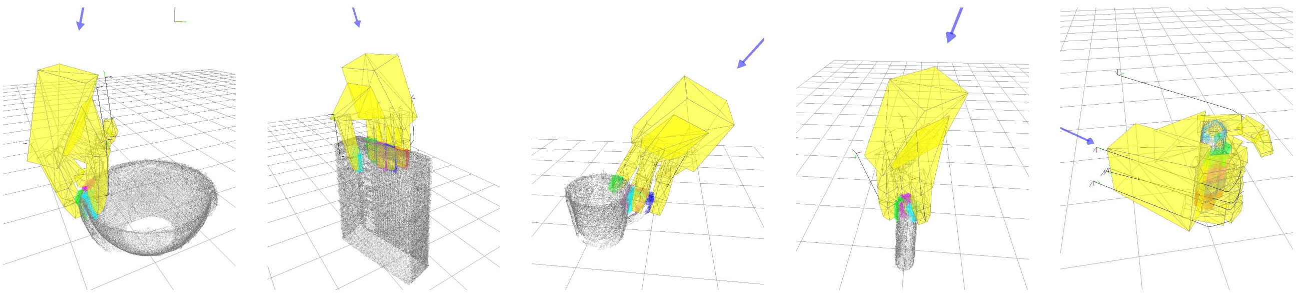


Fig. 8. The five training grasps. From left to right these are *pinch with support*, *power-box*, *handle*, *pinch*, and *power-tube*. The grey lines show the sequence of finger tip poses on the demonstrated approach trajectory. The whole hand configuration is recorded for this whole approach trajectory. The initial pose and configuration we refer to as the pre-grasp position. For learning the contact models and the hand configuration model only the final hand pose (the yellow hand pose) is used. The point clouds are the result of registration of seven views with a wrist mounted depth camera taken during training.

$$\mathcal{L}^g(h) = \mathcal{L}^g_W(h)\mathcal{L}^g_C(h)\mathcal{L}^g_Q(h) \tag{24}$$

$$= W(h_w, h_c)C^g(h_c) \prod_{Q_i \in \mathcal{Q}^g} Q^g_i\left(k^{\text{for}}_i(h_w, h_c)\right) \tag{25}$$

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 (24) with

$$\|\mathcal{L}^g(h)\| = \mathcal{L}^g_W(h)\mathcal{L}^g_C(h)\left(\mathcal{L}^g_Q(h)\right)^{\frac{N^{\max}_Q}{N^g_Q}}. \tag{26}$$

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 Algorithm 2 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. The remaining best scoring hand pose $h*$ is then used to generate a collision free approach trajectory to the pre-grasp wrist pose (also determined by the training grasp – see Fig. 8) and the trajectory and grasp are executed.

During grasp optimisation and selection the evaluation of product (23) accounts for over 95% of the computation time during the entire grasp inference process. A single evaluation of product (23) has time complexity $\Omega(N_Q K_Q) + \Omega(d_{h_c} K_C)$ where d_{h_c} is the dimensionality of h_c. The time complexity can be reduced to $\Omega(N_Q \log K_Q) + \Omega(d_{h_c} \log K_C)$ using k-nearest neighbour search methods (Weber et al., 1998). However, because of the overhead associated with such search structures, this approach is only justified for large values of K_Q and K_C.

5. Experimental Method

The grasp transfer performance was studied by training five models with the grasps of Fig. 8, and testing on 45 grasps of 43 unfamiliar objects (two objects were presented in two different poses each). All the training and testing reported here was performed with the Boris robot platform depicted in Fig. 1.

Views	Kernels	Initial grasp candidates	Steps K	Selection steps	Selected %	Final grasp candidates	(T_1, T_K)
1	5,000	50,000	500	1, 50	10%	500	(1,0.1)
7	2,000	2,500	500	None	100%	2,500	(1,0.1)

Table 1: Algorithm parameterisation for the two experimental conditions (1 and 7 views).

5.1. Training

Training proceeded as follows. Each training object was placed on the table, and seven views of the object were taken using a depth camera (PrimeSense Carmine 1.09). The resulting view specific depth clouds were then combined to form a single point cloud model. Stitching the point clouds together is trivial, since we know the exact pose of the camera at each frame from the robot's forward kinematics. Each grasp was then demonstrated kinesthetically by a human operator. The whole hand pose was recorded at five points along the trajectory on the final approach from a pre-grasp position selected by the operator (see Fig. 8). The hand configuration model and the contact models for each finger segment were learned from the final configuration of this trajectory. The remaining four configurations on the approach trajectory are only used to interpolate the hand configuration during execution of the approach for the transferred grasps, and are not used in model learning or grasp inference. Only kinematic information was used during training, and no force sensing of contacts was recorded.

5.2. Testing

The testing phase proceeded as follows: An object was selected from the test set, and placed on the table. Two experimental conditions were tested, where either 1 or 7 views of the test object were taken using a depth camera. The simulated annealing procedure was run using the parameters in Tab. 1. In each case the final grasp candidates were ranked by likelihood, and pruned for kinematically infeasible grasps due to collisions with the table surface. The grasp selected was the first ranked grasp. The grasp was then executed on the robot using a PRM path planner with optimisations to reach the pre-grasp position (Kopicki, 2010), and using the generated grasp trajectory thereafter. The robot hand is a DLR-HIT2 hand, which uses active compliant control based on motor current sensing at 1 kHz. The success of the grasp was determined by whether the robot could raise the object and hold it for 10 seconds. This procedure was followed for all 45 test objects for both viewing conditions. In addition for seven objects under the seven view condition the first ranked grasp of the next best grasp *type* was also tested, and for one object the first ranked grasp of the third best grasp *type* was tested. This led to a total of 98 grasps being executed across the two conditions, of which 90 were the first choice grasps. Grasp generation took an average of 23 seconds for the 1 view condition, and 12 seconds for the 12 view condition on a Intel Core i7 4-core 2.6GHz processor. Query density computation took an average of 0.7 seconds and 0.23 seconds respectively.

5.3. Results

Tab. 2 shows the grasp transfer success rate. When 7 views were taken of the test object, of the 45 first choice test grasps made 38 were successful, and 7 failed, giving a success rate among first choice grasps of 84.4%. Of the 53 different grasps (45 first choice, 7 second choice and 1 third choice) executed 46 were successful giving a success rate among all grasps of 86.7%. At least one of the first or second choice grasp worked for 95.6% of objects. When only one view was taken of the test object the successfully executed first choice grasps fell to 35, i.e. 77.8%. Fig. 9 shows examples of successful grasps. Each image pair shows the object, the partial point cloud (red), the planned grasp (yellow), and the grasp executed.

Fig. 9. The test objects with a visualisation of some of the successful grasps. Each grasp is shown by a pair of images, with the visualisation of the planned grasp and the obtained point cloud on the left, and the actual grasp execution on the right. Those from the 7 view and 1 view conditions can easily be distinguished by the proportion of the object covered by the recovered point cloud.

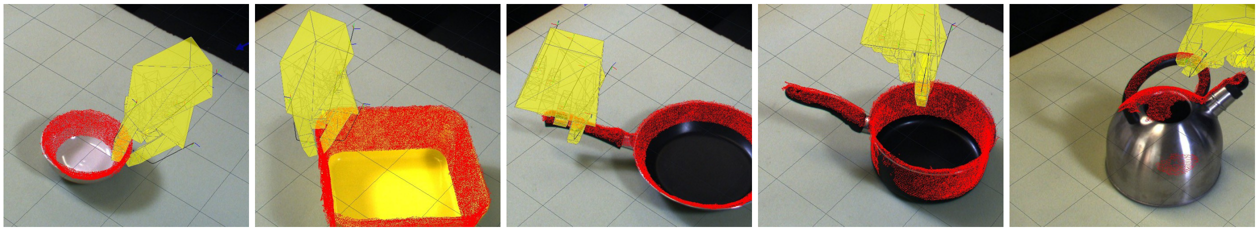


Fig. 10. The test objects with a visualisation of five of the failing grasps. The left four are from the 7 view condition and the kettle is from the 1 view condition.

		Absolute number (% of total) successes/failures			
Views	Testing objects/poses	1st choice successful	1st choice fail.	2nd & 3rd choice succ.	1st or 2nd choice succ.
1	45	35 (77.8%)	10 (22.2%)	n/a	n/a
7	45	38 (84.4%)	7 (15.6%)	8 (100%)	43 (95.6%)

Table 2: Grasp success rates for the two conditions.

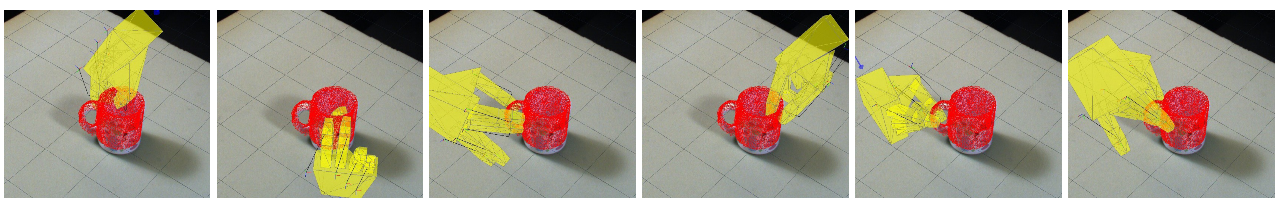


Fig. 11. Grasp variation on mug3. The grasp types used to generate these were (from top to bottom, left to right): pinch, pinch, pinch, pinch with support, handle and handle.

6. Discussion

There are several properties of the grasps generated worthy of further discussion. A complete set of images for all grasps is given in Extension 1. A supporting video with results is given in Extension 2.

Variety of grasp types. For several objects at least two grasp types were tried and executed successfully. A good example is given in Fig. 11, where the mug has six quite different grasps shown from the ranked set. Other examples from Fig. 9 include the pinch and power grasps on a coke bottle, and pinch and handle grasps on a cup. This shows the variety of grasp types the method is able to use when the object is presented in the same orientation. This variety matters for general grasping ability. In the seven view condition, when first grasps failed, the second choice grasp-type always succeeded (except two cases where it wasn't attempted for safety reasons). This means that from first or second choice grasp-types at least one of

Grasp type	1st choice occurrences	
	7 views	1 view
pinch	20	20
pinch w/ support	20	20
handle	3	4
power-box	2	0
power-tube	0	1

Table 3: First choice grasp type distribution for the two conditions.

these worked for 43 of the 45 object-pose combinations. Thus 95.5% of the test objects were successfully grasped by one of the top two grasp types.

Robustness to partial surface data. Fig. 9 shows the recovered point cloud for each test grasp. In the right column it can be seen that successful grasps were made in the face of quite small amounts of surface data being recovered. This is true even of quite complex grasps, such as the handle grasp of the mug.

Robustness to object pose. The method is robust to reorientations of the object. In Fig. 9 the large funnel is presented point up and bowl up respectively, and yet the grasp is adapted from the same base grasp (pinch with support) in both cases. In the case of the guttering the grasps selected are pinch and pinch with support respectively in response to different orientations of the guttering on the table.

Preference for simple grasps. The method is capable of generating a variety of grasp types, but the preferred grasps typically involve fewer finger links. This reflects the greater ease adapting them to more closely match the conditions of the original grasp: fewer finger links involved in the grasp means fewer constraints. This is a different property to the need to rescale grasp likelihood by the number of links involved. In that case as the number of links rises the grasp likelihood falls. That would be the case whether or not two grasps being compared were identical to their training grasps and evaluated on the training objects.

Grasping different object parts. The method generates a large number of grasps. These have a high degree of variation in their pose on the object. This is also shown in Fig. 11. Note that due to the incomplete point cloud some are reasonable even though they are not feasible. Since many missing points are underneath the object these grasps are typically not kinematically feasible either and so are pruned. The variety of grasps generated supports the idea that the method will allow grasping in cluttered scenes, or to find a suitable grasp in the face of task constraints, although testing these hypotheses falls beyond the scope of this paper.

Degree of generalisation. When viewing the transferred grasps next to the example grasp the degree of generalisation is notable. Fig. 12(top) shows three grasps together with the training grasps from which they were adapted. The adaptation from bowl to funnel spout and to a spray bottle shows the generalisation ability of the pinch with support grasp. The grasp of the guttering using a pinch grasp widens the finger spacings significantly with respect to the example grasp on the tube. In addition the global shape of these test objects is different from the training examples. The variety of grasps achievable by adapting one learned grasp is shown by the adaptations of the pinch with support grasp type in Fig. 9. The bucket, funnel (both orientations), guttering, kettle and spray bottle are all adaptations of this grasp type.

Failing grasps. it is worth analysing why grasps fail. Fig. 10 show five failing first choice grasps. The grasp of the bowl failed because the pinch grasp together with the low frictional coefficient of the objects don't give sufficient frictional contact to achieve force closure. In the case of the saucepan the grasp is in the wrong place: the grasp of the rim can't resist the wrench given by the large, heavy object, a grasp around the handle would be better. This was tried for the frying pan, but the wrong type of grasp was used. Instead an adaptation of the power-tube grasp succeeded on the saucepan in Fig. 9. The grasp of the kettle failed because in the single view condition the surface reconstruction is so limited it affects the grasp quality significantly. Finally some grasps, such as the grasp of the yellow container, fail while being superficially very similar to successful grasps of the same object.

7. Conclusions

This paper has presented a method that generalises a single kinesthetically demonstrated grasp to generate many grasps of other objects of different and unfamiliar shapes. One essential element is learning a separate *contact model* for each finger phalange how of its pose relative to the surface is related to local surface feature. This encodes local contact constraints. Another is learning a *hand configuration model* based on sampling poses near to those on the approach trajectory in the training example. This encodes the global handshape.

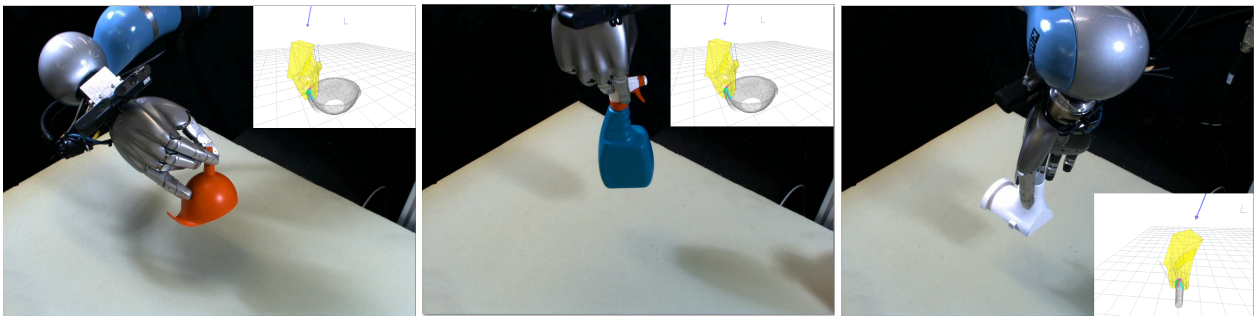


Fig. 12. Examples of transfer to globally very different objects.

The optimisation process is seeded by these two models with many different starting positions, and the resulting optimised grasps are then clustered to yield a variety of different viable grasp candidates. This is advantageous because the candidates can then be further assessed and ranked by stability, reachability, and task suitability. This paper has described simple ranking by similarity. This step could itself be a precursor to other analysis of the ranked grasps.

The empirical studies performed show that: i) the method can learn from one example grasp of a particular type; ii) the system creates grasps for objects with globally different shapes from the training objects; iii) for each new object many different new grasps can be generated, ordered by likelihood, 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) grasp success rates on test objects are high and robust to partial surface recovery: 84.4% with seven views, and 77.8% with one view.

7.1. Future Work

Many problems remain in dexterous grasping. This work provides a step forward in terms of grasp generation and generalisation. Reasoning about such grasps by other algorithms is the next step. In particular the appeal of dexterous hands is that they enable a variety of ways for the hand to interact with the object, and selecting the initial grasp so as to enable the particular chosen interactions or task is a necessary problem to tackle. Grasp refinement using reinforcement learning is another obvious route to improving the grasps created using the methods here.

References

C. Bard & J. Troccaz (1990). 'Automatic preshaping for a dextrous hand from a simple description of objects'. In *International Workshop on Intelligent Robots and Systems*, pp. 865–872. IEEE.

Y. Bekiroglu, et al. (2011). 'Integrating Grasp Planning with Online Stability Assessment using Tactile Sensing'. In *International Conference on Robotics and Automation*, pp. 4750–4755. IEEE.

H. Ben Amor, et al. (2012). 'Generalization of human grasping for multi-fingered robot hands'. In *International Conference on Intelligent Robots and Systems*, pp. 2043–2050. IEEE.

A. Bicchi & V. Kumar (2000). 'Robotic grasping and contact: a review'. In *International Conference on Robotics and Automation*, pp. 348–353. IEEE.

E. Borra, et al. (2011). 'Anatomical evidence for the involvement of the macaque ventrolateral prefrontal area 12r in controlling goal-directed actions'. *The Journal of Neuroscience* **31**(34):12351–12363.

G. I. Boutselis, et al. (2014). 'Task Specific Robust Grasping For Multifingered Robot Hands'. In *International Conference on Robotics and Automation*, pp. 858–863. IEEE.

J. Coelho, et al. (2000). 'Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot'. In *Robotics and Autonomous Systems*, vol. 37, pp. 7–8.

R. Detry, et al. (2013). 'Learning a Dictionary of Prototypical Grasp-predicting Parts from Grasping Experience'. In *International Conference on Robotics and Automation*, pp. 601–608. IEEE.

A. H. Fagg & M. A. Arbib (1998). 'Modeling parietal-premotor interactions in primate control of grasping'. *Neural Networks* **11**(7-8):1277–1303.

C. Ferrari & J. Canny (1992). 'Planning optimal grasps'. In *International Conference on Robotics and Automation*, pp. 2290–2295.

D. Fischinger & M. Vincze (2012). 'Empty the basket – a shape based learning approach for grasping piles of unknown objects'. In *International Conference on Intelligent Robots and Systems*, pp. 2051–2057. IEEE/RSJ.

R. A. Fisher (1953). 'Dispersion on a sphere'. In *Proc. Roy. Soc. London Ser. A.*, vol. 217, pp. 295–305. Royal Society.

A. K. Goins, et al. (2014). 'Evaluating the Efficacy of Grasp Metrics for Utilization in a Gaussian Process-Based Grasp Predictor'. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3353–3360. IEEE/RSJ.

I. Gori, et al. (2014). 'Three-Finger Precision Grasp on Incomplete 3D Point Clouds'. In *IEEE International Conference on Robotics and Automation*, pp. 5366–5373. IEEE.

R. Grupen (1991). 'Planning grasp strategies for multifingered robot hands'. In *IEEE International Conference on Robotics and Automation*, pp. 646–651. IEEE.

K. Hang, et al. (2014). 'Combinatorial Optimization for Hierarchical Contact-level Grasping'. In *IEEE International Conference on Robotics and Automation*, pp. 381–388. IEEE.

A. Herzog, et al. (2014). 'Learning of grasp selection based on shape-templates'. *Autonomous Robots* **36**(1-2):51–65.

U. Hillenbrand & M. Roa (2012). 'Transferring functional grasps through contact warping and local replanning'. In *IEEE/RSJ International Conference on Robotics and Systems*, pp. 2963–2970. IEEE.

Y. Hu, et al. (1999). 'Human Visual Servoing for Reaching and Grasping: The Role of 3-D Geometric Features'. In *IEEE International Conference on Robotics and Automation*, pp. 3209–3216. IEEE.

L. S. Jakobson & M. A. Goodale (1991). 'Factors affecting higher-order movement planning: a kinematic analysis of human prehension'. *Experimental Brain Research* **86**(1):199–208.

I. Kamon, et al. (1996). 'Learning to grasp using visual information'. In *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2470–2476.

K. Kanatani (2005). *Statistical optimization for geometric computation: theory and practice*. Courier Dover Publications.

J. Kim (2007). 'Example-based Grasp Adaptation'. Master's thesis, Massachusetts Institute of Technology.

J. Kim, et al. (2013). 'Physically Based Grasp Quality Evaluation Under Pose Uncertainty'. *IEEE Transactions on Robotics* **29**(6):1424 – 1439.

S. Kirkpatrick, et al. (1983). 'Optimization by simulated annealing'. *Science* **220**(4598):671–680.

E. Klingbeil, et al. (2011). 'Grasping with application to an autonomous checkout robot'. In *IEEE International Conference on Robotics and Automation*, pp. 2837–2844. IEEE.

G. W. Kootstra, et al. (2012). 'Enabling grasping of unknown objects through a synergistic use of edge and surface information'. *The International Journal of Robotics Research* **34**:26–42.

M. Kopicki (2010). *Prediction learning in robotic manipulation*. Ph.D. thesis, University of Birmingham.

M. Kopicki, et al. (2014). 'Learning dextrous grasps that generalise to novel objects by combining hand and contact models'. In *IEEE International Conference on Robotics and Automation*, pp. 5358–5365. IEEE.

O. Kroemer, et al. (2010). 'Combining Active Learning and Reactive Control for Robot Grasping'. *Robotics and Autonomous Systems* **58**(9):1105–1116.

O. Kroemer, et al. (2012). 'A kernel-based approach to direct action perception'. In *IEEE International Conference on Robotics and Automation*, pp. 2605–2610. IEEE.

Y.-H. Liu (2000). 'Computing n-finger form-closure grasps on polygonal objects'. *The International Journal of Robotics Research* **19**(2):149–158.

A. Morales, et al. (2004). 'Using Experience for Assessing Grasp Reliability'. *International Journal of Humanoid Robotics* **1**(4):671–691.

R. Platt, et al. (2006). 'Learning Grasp Context Distinctions that Generalize'. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pp. 504 – 511. IEEE.

N. S. Pollard (2004). 'Closure and quality equivalence for efficient synthesis of grasps from examples'. *The International Journal of Robotics Research* **23**(6):595–613.

J. Ponce & B. Faverjon (1995). 'On computing three-finger force-closure grasps of polygonal objects'. *IEEE Transactions on Robotics and Automation* **11**(6):868–881.

M. Popović, et al. (2010). 'A Strategy for Grasping Unknown Objects based on Co-Planarity and Colour Information'. *Robotics and Autonomous Systems* **58**(5):551–565.

M. Richtsfeld & M. Zillich (2008). 'Grasping unknown objects based on 2.5D range data'. In *IEEE International Conference on Automation Science and Engineering*, pp. 691–696. IEEE.

G. Rizzolatti & G. Luppino (2001). 'The cortical motor system'. *Neuron* **31**(6):889–901.

C. Rosales, et al. (2011). 'Synthesizing grasp configurations with specified contact regions'. *The International Journal of Robotics Research* **30**(4):431–443.

C. Rosales, et al. (2012). 'On the synthesis of feasible and prehensile robotic grasps'. In *IEEE International Conference on Robotics and Automation*, pp. 550–556. IEEE.

R. B. Rusu, et al. (2009). 'Perception for Mobile Manipulation and Grasping using Active Stereo'. In *IEEE-RAS International Conference on Humanoids*, pp. 632–638. IEEE.

J. Saut & D. Sidobre (2012). 'Efficient models for grasp planning with a multi-fingered hand'. *Robotics and Autonomous Systems* **60**(3):347–357.

A. Saxena, et al. (2008). 'Learning grasp strategies with partial shape information'. In *Proceedings of AAAI*, pp. 1491–1494. AAAI.

A. Shapiro, et al. (2004). 'On the mechanics of natural compliance in frictional contacts and its effect on grasp stiffness and stability'. In *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1264–1269. IEEE.

K. B. Shimoga (1996). 'Robot grasp synthesis algorithms: A survey'. *The International Journal of Robotics Research* **15**(3):230.

B. W. Silverman (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC.

M. Spivak (1999). *A comprehensive introduction to differential geometry*, vol. 1. Publish or Perish Berkeley.

E. B. Sudderth (2006). *Graphical models for visual object recognition and tracking*. Ph.D. thesis, MIT, Cambridge, MA.

A. ten Pas & R. Platt (2014). 'Localizing Handle-like Grasp Affordances in 3D Point Clouds'. In *International Symposium on Experimental Robotics*.

M. Trobina & A. Leonardis (1995). 'Grasping arbitrarily shaped 3-D objects from a pile'. In *IEEE International Conference on Robotics and Automation*, pp. 241–246. IEEE.

R. Weber, et al. (1998). 'A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces'. In *Proceedings of the 24th VLDB Conference*, vol. 98, pp. 194–205.

J. Xu, et al. (2007). 'Force analysis of whole hand grasp by multifingered robotic hand'. In *IEEE International Conference on Robotics and Automation*, pp. 211–216. IEEE.

L. E. Zhang, et al. (2011). 'Grasp Evaluation with Graspable Feature Matching'. In *RSS 2011 Workshop on Mobile Manipulation: Learning to Manipulate*.

Y. Zheng & W.-H. Qian (2005). 'Coping with the grasping uncertainties in force-closure analysis'. *The International Journal of Robotics Research* **24**(4):311–327.

Appendix A: Index to Multimedia Extensions

Extension	Media Type	Description
1	Images	Images of every grasp for both conditions (1 and 7 view)
2	Video	Video with high level explanation and results