Unsupervised Learning Of Predictive Parts For Cross-object Grasp Transfer

Renaud Detry and Justus Piater

***Abstract*— We present a principled solution to the problem of transferring grasps across objects. Our approach identifies, through autonomous exploration, the size and shape of object parts that consistently predict the applicability of a grasp across multiple objects. The robot can then use these parts to plan grasps onto novel objects. By contrast to most recent methods, we aim to solve the part-learning problem without the help of a human teacher. The robot collects training data autonomously by exploring different grasps on its own. The core principle of our approach is an intensive encoding of low-level sensorimotor uncertainty with probabilistic models, which allows the robot to generalize the noisy autonomously-generated grasps. Object shape, which is our main cue for predicting grasps, is encoded with surface densities, that model the spatial distribution of points that belong to an object's surface. Grasp parameters are modeled with grasp densities, that correspond to the spatial distribution of object-relative gripper poses that lead to a grasp. The size and shape of grasp-predicting parts are identified by sampling the cross-object correlation of local shape and grasp parameters. We approximate sampling and integrals via Monte Carlo methods to make our computer implementation tractable. We demonstrate the applicability of our method in simulation. A proof of concept on a real robot is also provided.**

I. Introduction

This paper addresses the problem of transferring grasps across objects, to the end of decreasing the cost of building grasp models for the novel objects that a robot encounters. Transferring grasps across objects is crucial for service robots, because object grasp models are expensive to construct. Constructing a grasp model with force analysis [4] is computationally expensive, and it requires the robot to spend time and energy in sensing, in order to make an accurate estimation of the object's shape and mass distribution. Learning grasp models from experience [10], [16], [27] is also time-consuming, as it requires the robot to try different grasps and to evaluate their workability.

Transferring grasps across objects has the potential to dramatically decrease the time required to construct grasp models for novel objects, as it provides the robot with a valuable prior on graspablity. In this paper, we suggest to exploit cross-object part redundancy to transfer grasps across objects that share similar parts.

While it is possible to implement part-based grasping by hard-coding a set of shape primitives (spheres, cones, boxes) and their associated grasps [26], it is becoming increasingly clear that, to work in an open-ended environment, robots need to adapt. In the context of part-based grasping, we want the robot to learn how to grasp parts that are frequently observed among the objects that it commonly works with. The idea of part-based grasp learning has already been tested by different research groups [1], [9], [18], [24], [33], [37]. A common factor of these methods is their reliance on human supervision to train the robot. The human teacher communicates to the robot his lifelong experience in grasping objects by constituent parts, by specifying which parts are often good for grasping, and in which way. As the teacher selects highly informative grasps, one or two examples per object are usually sufficient. As a result, supervised part-learning algorithms typically train on a large set of objects, with a couple of examples per object.

While human supervision does allow robots to learn at a remarkably fast pace, it is not always available. In cases where a robot needs to progress on its own, grasp learning is done via autonomous exploration [10], where the robot dedicates a slot of its time to testing different grasps suggested by visual cues [29]. Autonomous exploration produces a large number of grasps, that are on average of a lower stability and generality than grasps that are demonstrated by a teacher.

In this paper, we present a robotic agent that learns grasp-predicting parts from grasps collected via autonomous exploration. The robot thus learns on its own, without the help of a teacher. By contrast to supervised part-learning, where the teacher provides a few well-placed grasps on each object, this paper focuses on learning from training data composed of hundreds of grasping examples, where one cannot assume that all grasps are applied onto interesting parts. Because of the cost of autonomous exploration, the number of objects in our training database is smaller than the number of objects available to supervised-learning studies. In other words, we propose a part-learning algorithm that focuses on extracting a maximum amount of information from dense grasp model, to let promising object parts emerge from few but densely-annotated objects.

We assume the existence of dense grasp models that encode (1) the shape of an object, via a point cloud or similar representation, and (2) a set of object-relative wrist poses that lead to a grasp when the fingers are closed simultaneously until contact. The core principle of our approach is an intensive encoding of low-level visuomotor uncertainty via probabilistic models. The shape of objects (and of object parts) is encoded with surface densities [11], that model the spatial distribution of points that belong to an object's surface. Grasping parameters are modeled with grasp densities [10], that correspond to the spatial distribution of object-relative gripper poses that lead to a

R. Detry is with the University of Liège, Belgium. Email: renaud.detry@ulg.ac.be. J. Piater is with the University of Innsbruck, Austria. This work was supported by the Swedish Research Council (VR) and the Belgian National Fund for Scientific Research (FNRS).

grasp. This design choice allows us to smooth out the substantial amount of sensorimotor uncertainty that stems, first, from the noise associated to the real world, but also from local part/grasp variations from one object to the next. The idea behind our approach is to search through those models for object parts that afford similar grasps across different objects. The size and shape of grasp-predicting parts are identified by sampling the cross-object correlation of local shape and grasp parameters, over parts of different spatial extents. In essence, our approach hypothesizes parts of different sizes and shapes from the training models, by randomly segmenting pieces of the surface densities and grasp densities of the training objects. It then evaluates whether similar combinations of shape and grasps exist in the training set. Computing similarities between densities involves integrals that we cannot solve analytically. Instead, our method relies on Monte Carlo integral approximations to make our implementation tractable.

We demonstrate the applicability of our method in simulation and on a physical robot.

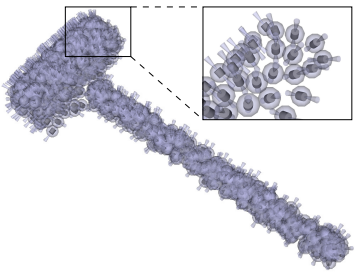


Fig. 1: Surface density computed from the 3D scan of a mallet. Surface points and their normals are rendered with cylinders. The axis of a cylinder represents the orientation of the local surface normal. Kernels are illustrated with translucent shapes: spheres and cones show one standard deviation in position and orientation respectively.

Fig. 2: Projection of a 6D grasp density onto a 2D image. For more details we refer the reader to the work of Detry et al. [10].

II. Related Work

In robotics, mainstream grasp planning has traditionally relied on force analysis [4]. Given a model of the shape, weight distribution, and surface texture of an object, and a model of the shape, kinematics, and applicable forces/torques of a gripper, force analysis allows us to compute the magnitude of the strongest external disturbance that a grasp can withhold. Force analysis is applicable to multi-fingered hands, and its ability to generate complex grasps has been shown in the literature [17], [32], [36]. The application of force analysis to grasping novel objects has been studied, for instance through the construction of object shape models from noisy and incomplete sensor data, and the use of heuristics to define mass distribution and friction parameters [31]. Unfortunately, the strengths of force analysis become mitigated in scenarios where the object models that the robot can recover (either from memory or sensor data) are incomplete or lack in accuracy.

A number of authors have explored means of directly linking perceptions to action parameters. Authors have used symmetry principles to reconstruct the occluded side of unknown objects [35], and therefore allow the definition of grasp contact points on occluded surfaces [5], [20]. Other groups have developed means of parameterizing grasps by looking for shapes that are likely to fit into the robot's gripper [14], [22], [29]. Popovic et al. [29] computed grasps applied to object edges detected in 2D images. Another class of methods searched 3D range data for shapes that closely match the geometry of the gripper [14], [22].

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 [8], [21], [28].

Goldfeder et al. [16] have presented a data-driven approach, where a large number of grasps are conducted in simulation, and grasp parameters for a novel object A are recovered by selecting from a training database the object that best matches A's shape. As the size of the database increases, the likelihood of finding an object similar to A grows. Unfortunately, the time required to find a match also increases with the number of training objects. To alleviate this problem, authors have searched for means of transferring grasps between object *parts*. Ben Amor et al. [2] and Hillenbrand et al. [19] aimed at transferring grasps across objects whose complete 3D shape is known. Closer to our work, several authors focused on learning a mapping from incomplete object views to grasp parameters [1], [9], [18], [24], [23], [33], [37]. Five of these approaches focus on the same problem as ours, i.e., learning the 3D shape of graspable parts [1], [9], [18], [24], [37]. All of these approaches rely on human supervision, and thus assume that all the grasps present in the training data are applied onto a part that has a high likelihood of being shared across multiple objects. As explained above, our work focuses instead on denser training data, where one cannot assume that all grasps are applied onto interesting parts.

III. Probabilistic Models

Our method relies on probabilistic models of low-level sensory data, where probability density functions (PDFs) play an important role. To avoid clutter, instead of using the standard $p()$ notation to denote PDFs, we denote PDFs with any lowercase bold letter, such as $\mathbf{p}()$, $\mathbf{v}()$, or $\mathbf{g}()$. The notation $\hat{x} \sim \mathbf{p}(x)$ is used to denote an element $\hat{x}$ sampled from $\mathbf{p}(x)$. The probabilistic models that our work relies on are briefly introduced below.

A. Surface Densities

We use surface densities [11] to model the 3D shape of object parts and of whole objects. Let $\mathcal{V} = \{a_i\}_{i\in[1,n]}$ denote the point-cloud representation of an object (or an object part), with all a_i's belonging to $\mathbb{R}^3$, and let $\mathcal{V}' = \{a'_i\}_{i\in[1,n]}$ be

the set of surface normals computed at each a_i. The surface density of $\mathcal{V}$, denoted by $\mathbf{v}(y)$, is constructed via kernel density estimation (KDE) [34] on the set $\{(a_i, a'_i)\}_{i\in[1,n]}$, by centering a kernel function onto each datapoint, and summing the kernels. As a result, $\mathbf{v}(y)$ is defined on the product space of 3D positions and surface normals $\mathbb{R}^3 \times S^2$. The kernel function supporting KDE is defined as the product of a trivariate Gaussian and a two-sphere von Mises–Fisher distribution [15]. Intuitively, the value of $\mathbf{v}(y)$ at a given point $y \in \mathbb{R}^3 \times S^2$ is inversely proportional to the distance between y and its closest neighbors in $\mathcal{V}$. Fig. 1 illustrates surface densities. For further details on this model, we refer the reader to the work of Detry et al. [11].

B. Pose Estimation

Let $t_x(\cdot)$ denote a rigid transformation by $x \in SE(3)$, and let $t_x^{-1}(\cdot)$ denote the inverse of $t_x(\cdot)$, such that

$$(t_x \circ t_x^{-1})(y) = y \tag{1}$$

for all y in $SE(3)$ or $\mathbb{R}^3 \times S^2$. For clarity, in the equations below, $t_x(y)$, which gives the transformation of y by x, will be denoted by $y + x$. Similarly, $t_x^{-1}(y)$ will be denoted by $y - x$. Let us consider an object whose point cloud is denoted by $\mathcal{V}$, and whose surface density is denoted by $\mathbf{v}(y)$, and let us consider $\mathcal{W}$, a partial view of a scene where the same object appears, whose surface density is denoted by $\mathbf{w}(y)$.

Our probabilistic setting allows us to formulate an elegant solution to the pose estimation problem, by providing us with means of computing the *pose density* $\mathbf{p}(x)$ of the object. The pose density is a PDF that models the probability of the object standing at any given $SE(3)$ pose x. The pose density of the object $\mathcal{V}$ above can be expressed by marginalizing the joint distribution of object poses and visual observations, as

$$\mathbf{p}(x) = \int \mathbf{p}(x|y)\mathbf{w}(y)\mathrm{d}y. \tag{2}$$

In the equation above, the conditional pose probability $\mathbf{p}(x|y)$ is simply given by

$$\mathbf{p}(x|y) = \mathbf{v}(y - x). \tag{3}$$

Intuitively, for a given x, $\mathbf{p}(x|y)$ is equal to the PDF $\mathbf{v}(y)$ translated and rotated by x. The probability of $\mathcal{V}$ being at pose x is then given by Eq. 2, which will give a value proportional to the overlap between $\mathbf{p}(x|y)$ and $\mathbf{w}(y)$. In other words, Eq. 2 is the $SE(3)$ cross-correlation of $\mathbf{v}$ and $\mathbf{w}$.

Eq. 2 above is not tractable analytically. Instead, we approximate it with Monte Carlo integration [6], [11], as

$$\mathbf{p}(x) \simeq \frac{1}{M}\sum_{\ell=1}^{M} \mathbf{p}(x|y_\ell) \quad \text{where} \quad y_\ell \sim \mathbf{w}(y), \tag{4}$$

where M is a large numerical constant.

C. Grasp Densities

In this work, grasps are parametrized by the 6D pose (3D position and 3D orientation) of the gripper. We use grasp densities (GDs) [10] to model the grasps afforded by an object, or by an object part. GDs model the different ways to place a hand or a gripper near the object so that closing the gripper produces a stable grip. Specifically, GDs encode object-relative gripper configurations and the probability of their grasping success. GDs are mathematically close to the surface densities discussed above. Denoting by $\mathcal{G} = \{b_i\}_{i\in[1,m]}$ a set of object-relative grasp examples, with all b_i's belonging to $\mathbb{R}^3 \times SO(3)$, The grasp density $\mathbf{g}(x)$ is constructed via kernel density estimation, by centering a kernel function onto each input datapoint, and summing the kernels. The kernel function is defined as the product of a trivariate Gaussian and a three-sphere von Mises–Fisher distribution [15], [10].

D. Object Model

In the rest of the paper, we call *object model* the association of a surface density modeling the shape of an object, and a grasp density modeling object-relative wrist poses that lead to a grasp. We denote the surface density of an object o by $\mathbf{v}_o$, and its grasp density by $\mathbf{g}_o$. The training database, composed of N objects, is denoted by

$$L = \left\{o^{(i)}\right\}_{i\in[1,N]} \qquad \text{with } o^{(i)} = (\mathbf{v}_{o^{(i)}}, \mathbf{g}_{o^{(i)}}). \tag{5}$$

IV. Recurring Visuomotor Parts

Let us define a *visuomotor part* p as the association of a surface density $\mathbf{v}_p$ modeling an object part, and a grasp density $\mathbf{g}_p$ explaining how to grasp that part. In order to allow the agent to generalize its grasping knowledge to new objects, we suggest to let the agent search for *recurring* visuomotor parts across the set of previously-acquired models, counting on the fact that parts that are observed multiple times across multiple objects are likely to be applicable to novel objects.

To this end, we define a measure of *part generality*. The generality of a part is measured by its ability to predict grasps across the library L of known object models. This measure relies on a function $f(p, o)$ that yields a high value if the visual component of a part p successfully identifies regions of an object o associated to grasping strategies that are similar to its own, and discards those that are not. The generality measure of p is defined from the statistics of $\{f(p, o) : o \in L\}$. As described below, the agent will systematically evaluate the generality of parts randomly segmented from existing models, yielding a set of parts ordered by their ability to generalize. Those that yield a high measure will be selected to grasp new objects. We note that the process of discovering recurring parts (i.e., generalization) is run offline – it is entirely based on previously-acquired object models (the training models) and it does not require the robot to execute grasps.

The object model defined above offers efficient and elegant means of implementing f. Its probabilistic representation of visual structure and grasp strategies with density functions

defines a convenient abstraction which allows us to think of solutions in terms of generic statistics and machine-learning tools. In particular, we detail below how the grasping correlation between a visuomotor part and an object model can be approximated with the Bhattacharyya distance [3].

As described above, our goal is to identify a set of visuomotor parts for which the visual component is a robust predictor of the grasping component. These generic parts are discovered as follows:

1) Randomly segment a set of P object parts $\left\{p^{(i)}\right\}_{i\in[1,P]}$ from the library of known objects $L = \left\{o^{(i)}\right\}_{i\in[1,N]}$ (described below in Section IV-A).
2) For each part p, compute a generality measure $m(p, L)$ with respect to the set of known objects L (described below in Section IV-B).

The parts that yield a high measure will be selected for creating the initial grasp models of new objects (described below in Section IV-C).

A. Generating Candidates

Segmenting one object part p from the object library L (Eq. 5) works as follows:

1) Select one object model $o = (\mathbf{v}_o, \mathbf{g}_o)$ from L.
2) Select r uniformly in $[0, d]$, where d is the diameter of o's bounding sphere.
3) Let a be the position of a grasp randomly sampled from $\mathbf{g}_o$, and A correspond to the subset of the grasps that were used to build $\mathbf{g}_o$ that lie within a sphere of radius r centered at a.
4) The grasp density of p, denoted by $\mathbf{g}_p$, is defined as the KDE of A. The visual model of p, denoted by $\mathbf{v}_p$, is defined as the KDE of points of $\mathbf{v}_o$ which lie within the sphere of radius r centered at a.

B. Generality Measure

This section defines a measure $m(p, L)$ of the generality of part p with respect to the set of known objects L. We start by defining $f(p, o)$, i.e., the ability of p to predict the grasp model of a single object o. Let us denote p's model by $p = (\mathbf{v}_p, \mathbf{g}_p)$, and the object by $o = (\mathbf{v}_o, \mathbf{g}_o)$.

As described above (2), we can easily compute the pose probability of p in o, as

$$\mathbf{q}(x) = \int \mathbf{q}(x|y)\mathbf{v}_o(y)\mathrm{d}y, \quad \mathbf{q}(x|y) = \mathbf{v}_p(y - x). \tag{6}$$

The models $\mathbf{v}_p$, $\mathbf{g}_p$, and $\mathbf{v}_o$ allow us to definite of a grasp density $\mathbf{h}_o$ for the object modeled by o, as

$$\mathbf{h}_o(x) = \frac{1}{C}\int \mathbf{g}_p(x - z)\left[\mathbf{q}(z)\right]^c \mathrm{d}z, \tag{7}$$

where C is a normalizing factor, and c controls the trade-off between robust prediction and generalization. The expression $\mathbf{g}_p(x - z)$ corresponds to $\mathbf{g}_p(x)$ translated and rotated by z. Intuitively, the integral (7) considers all the different ways to align the part p with the object. The density $\mathbf{h}_o$ is computed as the weighted sum of all possible alignments of $\mathbf{g}_p$, where weights – given by $\mathbf{q}(z)$ – are computed from visual correlation. The constant c controls the trade-off between robust prediction and generalization. If $c = 0$, $\mathbf{h}_o$ represents random grasps. If $c = 1$, Eq. 7 is a standard marginalization. As c grows, $\mathbf{h}_o$ converges towards the transformation of $\mathbf{g}_p$ by $\arg\max_z \mathbf{q}(z)$, i.e., the transformation of $\mathbf{g}_p$ by the maximum-likelihood pose of $\mathbf{v}_p$ in $\mathbf{v}_o$. In the experiments below, c is set to 5.

The ability of p to predict the grasping properties of the object modeled by o can be measured by the similarity of $\mathbf{g}_o$ (the grasp density of o constructed empirically) and $\mathbf{h}_o$ (the grasp density of o constructed from p). Using the Bhattacharyya coefficient [3], this similarity is written as

$$f(p, o) = \int \sqrt{\mathbf{h}_o(x)\mathbf{g}_o(x)}\mathrm{d}x, \tag{8}$$

where $f(p, o) = 1$ if $\mathbf{h}_o(x) = \mathbf{g}_o(x)$ for all x. In our computer implementation, $f(p, o)$ is approximated via Monte Carlo integration. Noting that the expression of $f(p, o)$ can be rewritten as

$$f(p, o) = \int \sqrt{\frac{\mathbf{h}_o(x)\mathbf{g}_o(x)}{[\mathbf{g}_o(x)]^2}}\mathbf{g}_o(x)\mathrm{d}x, \tag{9}$$

the approximation is given by

$$f(p, o) \simeq \frac{1}{M}\sum_{\ell=1}^{M}\sqrt{\frac{\mathbf{h}_o(x_\ell)}{\mathbf{g}_o(x_\ell)}} \quad \text{where} \quad x_\ell \sim \mathbf{g}_o(x). \tag{10}$$

The generality of p with respect to the object library L is computed from the statistics of $\{f(p, o) : o \in L'\}$, where L' corresponds to L minus the object from which p was segmented. In the experiments below, the generality of p is computed as the arithmetic mean of $\{f(p, o) : o \in L'\}$

$$m(p, L) = \frac{1}{N - 1}\sum_{o\in L'} f(p, o). \tag{11}$$

C. Transferring Grasps to a Novel Object

The procedure described above is run offline to produce a large number of parts characterized by a generality measure. The k parts that generalize best are selected to form a dictionary that will allow the robot to grasp new objects. We denote by $K = \left\{p^{(i)}\right\}_{i\in[1,k]}$ the set of selected parts. Given the visual model $\mathbf{v}_{\hat{o}}$ of a novel object $\hat{o}$, our dictionary of parts allows us to compute a grasp density for $\hat{o}$, as

$$\mathbf{h}_{\hat{o}}(x) = \frac{1}{C}\int \sum_{p\in K} \mathbf{g}_p(x - z)\left[\mathbf{q}_p(z)\right]^c \mathrm{d}z, \tag{12}$$

where C is a normalizing factor, and $\mathbf{q}_p$ is computed from $\mathbf{v}_p$ and $\mathbf{v}_{\hat{o}}$ using Eq. 6. If the number of parts k is equal to 1, the expression above correspond to Eq. 7. If $k > 1$, $\mathbf{h}_{\hat{o}}(x)$ is constructed from a combinations of the parts in K, and the contribution of each part p is weighted by its shape resemblance with $\mathbf{v}_{\hat{o}}$. For instance, let us consider a set K containing two parts that model a cylinder ($p^{(1)}$) and a handle ($p^{(2)}$), and let $\mathbf{v}_{\hat{o}}$ correspond to a mug. Through the integral above (12), the region of $\mathbf{h}_{\hat{o}}(x)$ surrounding the cylinder of the mug will be computed from $p^{(1)}$ only, while the region surrounding the handle will be computed form $p^{(2)}$ only.

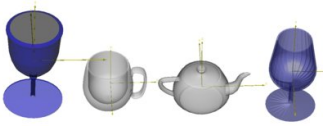


Fig. 3: Mesh models of the four objects used in this experiment: a goblet, a mug, a teapot, and a wine glass.

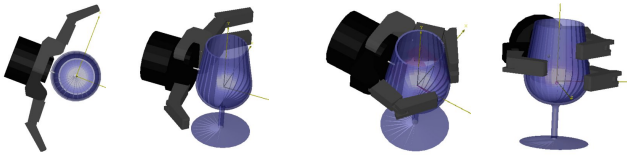


Fig. 4: Grasping an object in *GraspIt!*. According to the ϵ measure described in the text, the quality of this grasp is 0.24.

V. Experimental Results

This section presents two evaluations of our part generalization method. In Section V-A we evaluate our method in simulation, and we compare the success rate of a grasp planner that uses no prior grasping knowledge to the success rate of grasps transferred from other objects. The next section (Section V-B) illustrates the generality measure $f(p, o)$ on object models constructed with a real robot.

A. Generalization in a Simulated Environment

In this section, a robot learns part models in simulation, and we quantitatively evaluate their applicability to novel objects. The objects used in this experiment are presented in Fig. 3.

To prepare for this experiment, we first need to devise a means of acquiring grasp models for the training objects. Once the training models are available, we will learn a set of part models, and use the resulting parts for grasping a new object.

1) Surface-normal Grasp Planner: The training models are generated in simulation, by executing between 100000 and 150000 grasps. Grasps are planned as follows. Simulations are conducted in the *GraspIt!* simulator [25] using a Barrett hand model (Fig. 4). The simulated environment contains only the object and the hand. The hand is free-floating; the rest of the robot is not modeled. The simulator does not model dynamics: When closing the hand, fingers stop as soon as they make contact with the object. Grasp success is computed with the "ϵ" force-closure quality measure formulated by Ferrari and Canny [13]. The ϵ force-closure quality measure studies contact forces to characterize the effort the robot has to make to maintain force-closure under a worst-case external disturbance. The value of ϵ can vary between 0 and 1, with $\epsilon = 1$ corresponding to a very good grasp. In our experiment, a grasp is successful if $\epsilon > 0.05$.

The grasps from which our training models are generated are planned as follows. Let us denote by $\mathcal{V}$ the point cloud of the object. One point $a \in \mathcal{V}$ is selected at random, and the local surface normal is computed from its 16 nearest neighbors. The hand is moved towards a, fully open, with its palm perpendicular to the local surface normal. The rotation of the hand around the approach vector is selected from a uniform distribution on $[0, 2\pi[$. When the hand comes in contact with the object, the fingers are closed until they make contact, and the quality of the grasp is computed. KDE is applied to the set of grasps whose ϵ-quality is larger than 0.05 to produce a grasp density. In the rest of the paper, this way of computing grasps is referred to as the *surface-normal* grasp planner.

2) Learning Part Models: The rest of the experiment is organized as a leave-one-out cross-validation. Three of the four objects of Fig. 3 are used for learning part models. A grasp model is created for each of these three objects, following the procedure described in the previous section. The fourth object, let us denote it by o, is left out for testing. From the three training objects we generate a set of one hundred visuomotor parts (Section IV-A), and we compute their generality measure (Section IV-B). We then select the part with the highest generality measure, and we use it to construct a grasp density $\mathbf{g}_{\hat{o}}$ for the fourth object $\hat{o}$, using Eq. 12. (The set K of Eq. 12 thus contains a single part.) Fig. 7 shows a few grasps sampled from the resulting grasp density. We finally evaluate the applicability of our method by computing the success rate of grasps randomly sampled from $\mathbf{g}_{\hat{o}}$.

The process presented in the previous paragraph is repeated four times, to use each of the four objects for testing once. The four parts which present the highest generality measure across each subset of three objects are shown in Fig. 6.

Success rates are shown in Table I. We emphasize that these are success rates of grasps *randomly sampled* from the grasp density generated from a part model. This way, our evaluation characterizes the whole model constructed through part transfer, instead of characterizing a single grasping point.

As a comparison, Table II shows the success rate of grasps planned by the surface-normal planner discussed above. As shown in Table III, the success rates of the transfer-based planner are on average three times higher than those of the surface-normal planner.

In several respects, this experiment is rather simple, as objects share one obvious common part – a round-shaped "bowl", and grasp transfer is done through a single part model. Nonetheless, our results make the strengths of our approach explicit. The parts selected by the generality measure of Section IV-B seem intuitively pertinent. Their visual models include enough structure to encode the curvature of the underlying surface and correctly align grasps to similar shapes, while excluding structures that are not common to all objects.

B. Generalization with Models Learned by a Robot

This section illustrates the generality measure defined above on two real-world objects (see Fig. 8). In this section, visual models differ slightly from those discussed earlier.

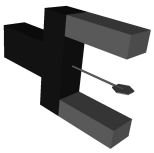
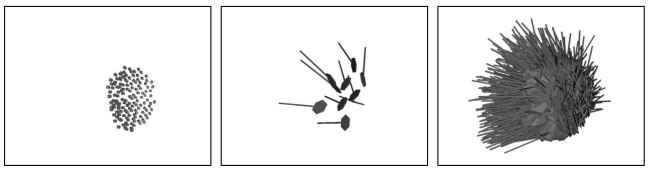
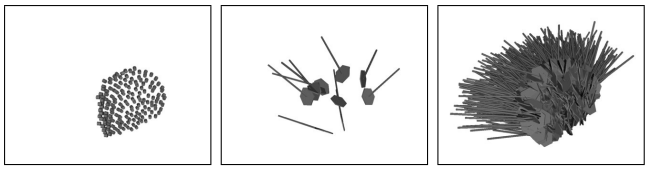


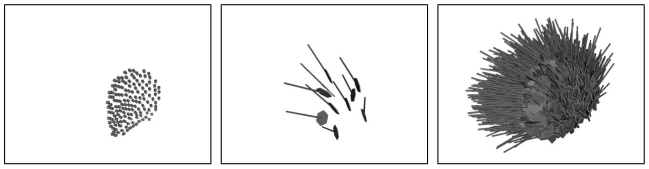
Fig. 5: For clarity, we render grasps with a small lollipop-like object. This image shows how it relates to a two-finger gripper.



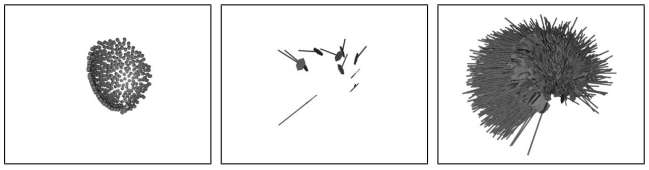
(a) Visuomotor part learned from the mug, the teapot, and the wine glass



(b) Visuomotor part learned from the goblet, the teapot, and the wine glass



(c) Visuomotor part learned from the goblet, the mug, and the wine glass



(d) Visuomotor part learned from the goblet, the mug, and the teapot

Fig. 6: Generic parts. Each triplet of images illustrates a part. The leftmost images correspond to the visual components. The middle and right-side images illustrate the grasp components. The right-side images show a large number of samples; the middle images show only ten. The four parts illustrated in this figure correspond to the parts of highest generality measure across each of the corresponding groups of three training objects. Fig. 5 illustrates how the grey "lollipops" relate to an actual gripper.

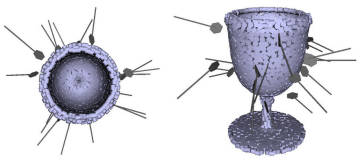


Fig. 7: Illustration of the generalization-based grasp density for the goblet. The figure shows in gray ten grasps sampled from the density created from the part in the top-row of Fig. 6.

Object	Successful Grasps	Tot. N. Grasps	Success Rate
Goblet	4948	23645	21%
Mug	5756	15191	37.9%
Teapot	4715	17712	26.6%
Wine Glass	3265	18267	17.9%

TABLE I: Success statistics of grasps transferred to a new object.

Object	Successful Grasps	Tot. N. Grasps	Success Rate
Goblet	9236	157282	5.9%
Mug	15119	119913	12.6%
Teapot	11453	136415	8.4%
Wine Glass	8096	126254	6.4%

TABLE II: Success statistics of the surface-normal grasp planner (see text for details).

	Goblet	**Mug**	**Teapot**	**Wine Glass**
Surface-normal planner	5.9%	12.6%	8.4%	6.4%
Grasp transfer	21%	37.9%	26.6%	17.9%

TABLE III: Success rates for the surface-normal planner and grasp transfer.

Fig. 8: Object library: toy pan an knife

Fig. 9: Visual models of the pan and knife [12], [30].

Instead of modeling the surface of objects with surface densities, we model object edges with edge densities [11], [12]. Those models are acquired from a stereo vision system that computes the 3D position and orientation of short edge segments extracted from the objects [30] (See Fig. 9). From a mathematical viewpoint, they are equivalent to the models discussed above, as each edge segment is parametrized by a 3D position and a 2D direction, as it is for surface points augmented with their surface normals.

The grasp models are learned through exploration on a real robot. More details on the acquisition of these models are available in our previous work [10]. The grasp models are illustrated in Fig. 10.

One hundred parts were randomly segmented from the model of the pan, and one hundred from the model of the knife. The ability of each part p of the pan to predict the

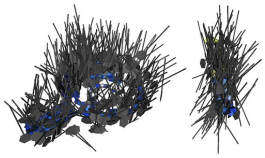
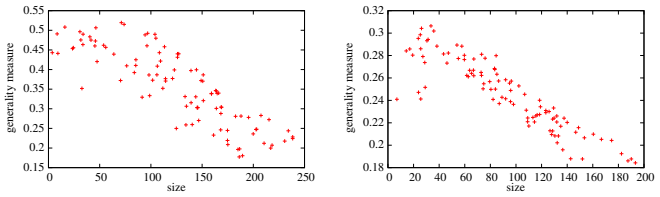


Fig. 10: Samples from empirical grasp densities learned with the objects of Fig. 8.



(a) Parts segmented from the pan (b) Parts segmented from the knife

Fig. 11: Generality measure of the parts of the pan with respect to the model of the knife (Fig. (a)) and of the parts of the knife with respect to the model of the pan (Fig. (b)). The measure of generality is plotted as a function of part sizes. See text for details.

empirical density of the knife was then computed using the generality measure $m(p, \{\text{knife}, \text{pan}\}) = f(p, \text{knife})$ defined above (11). Likewise, the generality measure of the pan model and each of the one hundred parts segmented from the knife was computed. An interesting way to plot these results is to show the generality measure as a function of the spatial size of the corresponding parts. In Fig. 11a, each cross corresponds to one of the hundred candidate parts segmented from the pan. The abscissa of a cross gives the diameter of the bounding sphere of the corresponding part. The ordinate of a cross gives the generality measure of the corresponding part. Fig. 11b shows a similar plot for parts segmented from the knife. Fig. 11 reveals one of the most important results of this paper: In the two plots of that figure, we can see that our algorithm makes it explicit that an optimal part size exists. As parts become smaller or larger than this optimum, the generality measure decreases. This result is intuitive, as the two objects we are considering have different shapes. When a part is too large, it does not resemble any subpart of the other object. One the other hand, when a part becomes smaller, it becomes too generic and fails to properly suggest grasps. Fig. 12 shows the knife part with the highest generality measure – it corresponds to the highest point of Fig. 11b. This part corresponds to a segment of about 1cm from the handle of the knife.

Fig. 13 shows three parts of the pan. The generality measure of the first two is high. The part of Fig. 13a is at coordinates $(71.2, 0.519)$ in Fig. 11a – its generality measure is 0.519, and the radius of its bounding sphere is 71.2mm. The part of Fig. 13b is at coordinates $(16, 0.508)$. By contrast, the part of Fig. 13c, which is rather large (186mm), has a low generality measure (0.177). These results correspond to what we would expect from a generality measure: the parts of Fig. 13a and Fig. 13b seem to contain information relevant to the knife, while the part of Fig. 13c could not be properly fitted to it.

VI. Discussion

One problem that is not addressed in this paper is the adaptation of the hand's fingers to an object's shape. Our work could however easily be extended to model finger preshapes or configurations along with the wrist position and orientation. While such a model would enable the generalization of dexterous grasps, it would also increase the size of the space that the robot needs to sample in order to learn the model parameters. Reducing the size of the finger configuration space [7] would greatly help, allowing for a continuous finger model, while limiting the number of additional latent dimensions.

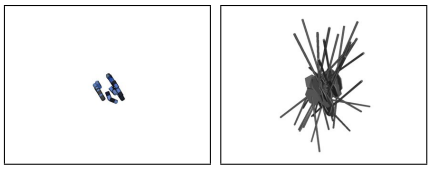


Fig. 12: Visuomotor model of a part segmented from the knife. The part corresponds to a small segment of the handle of the knife. The left image shows the visual component $\mathbf{v}_p$ of the part. The right image shows the associated grasp component $\mathbf{g}_p$.

Our C++ implementation running on a 2009 8-core Xeon computer produces the results discussed above in about two days. The memory footprint of the program is always below 50MB. The computational cost of ranking parts is $O(kN)$ where k is the number of parts that have to be tested, and N is the number of objects. From a computational viewpoint, linearity in the number of objects allows the model to scale to larger environments, as long as one manages to limit k. Currently, parts are generated in a random fashion that does not prevent similar parts from being tested (Section IV-A). Generating parts, and comparing them to one another, is computationally cheap by comparison to the evaluation of Eq. 7 and Eq. 8. Therefore, removing redundant parts before checking their generality measure could substantially contribute to scaling the method to larger environments.

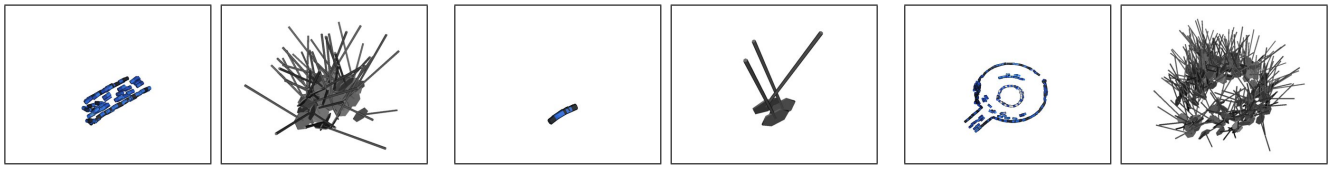
VII. Conclusions

We presented a principled solution to the problem of transferring grasps across objects. By contrast to methods that learn graspable parts from human demonstrations, we focused on an autonomous discovery of graspable parts from dense grasp models collected via exploratory learning. One key aspect of this work is that it does not assume that the grasps of the training database are all applied onto interesting parts. Instead, interesting parts emerge from a cross-object search for recurring visuomotor parts.

We presented a methodical approach to searching for recurring parts, that we based on an intensive encoding of low-level visuomotor uncertainty through surface densities and grasp densities. Similarities across objects are computed via probabilistic measures, which we approximated by Monte Carlo integration. We demonstrated the applicability of our approach in simulation and on two real-world objects.

VIII. Acknowledgements

The authors thank Prof. Jeremy Wyatt for his insights on formalizing our models. We thank Prof. Norbert Krüger for



(a) (b) (c)

Fig. 13: Visuomotor models of parts segmented from the pan. Fig. (a) corresponds to a small segment of the handle of the pan. Fig. (b) corresponds to an even smaller segment of the handle, while Fig. (c) is a model of almost all of the object. Images on the left show the visual component of each model, while images on the right show the grasp component.

his help with data collection. The *GraspIt!* simulator was developed by the Robotics Lab at Columbia University, NY.

REFERENCES

[1] C. Bard and J. Troccaz. Automatic preshaping for a dextrous hand from a simple description of objects. In *IEEE International Workshop on Intelligent Robots and Systems*, pages 865–872. IEEE, 1990.

[2] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters. Generalization of human grasping for multi-fingered robot hands. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 1943.

[4] A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE International Conference on Robotics and Automation*, 2000.

[5] 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 International Conference on Robotics and Automation*, pages 686–693, 2011.

[6] R. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[7] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *Int. J. Rob. Res.*, 28(7):851–867, 2009.

[8] 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.

[9] R. Detry, C. H. Ek, M. Madry, and D. Kragic. Learning a dictionary of prototypical grasp-predicting parts from grasping experience. In *IEEE International Conference on Robotics and Automation*, 2013.

[10] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger, and J. Piater. Learning grasp affordance densities. *Paladyn. Journal of Behavioral Robotics*, 2(1):1–17, 2011.

[11] R. Detry and J. Piater. Continuous surface-point distributions for 3D object pose estimation and recognition. In *Asian Conference on Computer Vision*, pages 572–585, 2010.

[12] R. Detry, N. Pugeault, and J. Piater. A probabilistic framework for 3D visual object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1790–1803, 2009.

[13] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

[14] D. Fischinger and M. Vincze. Empty the basket – a shape based learning approach for grasping piles of unknown objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[15] R. A. Fisher. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A.*, 1953.

[16] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen. The Columbia grasp database. In *IEEE International Conference on Robotics and Automation*, 2009.

[17] R. Grupen. Planning grasp strategies for multifingered robot hands. In *IEEE International Conference on Robotics and Automation*, 1991.

[18] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal. Template-based learning of grasp selection. In *IEEE International Conference on Robotics and Automation*, 2012.

[19] U. Hillenbrand and M. Roa. Transferring functional grasps through contact warping and local replanning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[20] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones. Contact-reactive grasping of objects with partial shape information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1228–1235, 2010.

[21] I. Kamon, T. Flash, and S. Edelman. Learning to grasp using visual information. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2470–2476, 1996.

[22] E. Klingbeil, D. Rao, B. Carpenter, V. Ganapathi, A. Ng, and O. Khatib. Grasping with application to an autonomous checkout robot. In *IEEE International Conference on Robotics and Automation*, 2011.

[23] O. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems*, 58(9):1105–1116, 2010.

[24] O. Kroemer, E. Ugur, E. Oztop, and J. Peters. A kernel-based approach to direct action perception. In *IEEE International Conference on Robotics and Automation*, 2012.

[25] A. Miller and P. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

[26] A. T. Miller, S. Knoop, H. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1824–1829, 2003.

[27] L. Montesano and M. Lopes. Learning grasping affordances from local visual descriptors. In *IEEE International Conference on Development and Learning*, 2009.

[28] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil. Using experience for assessing grasp reliability. *International Journal of Humanoid Robotics*, 1(4):671–691, 2004.

[29] 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. *Robotics and Autonomous Systems*, 2010.

[30] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. *International Journal of Humanoid Robotics*, 2010.

[31] R. B. Rusu, A. Holzbach, R. Diankov, G. Bradski, and M. Beetz. Perception for mobile manipulation and grasping using active stereo. In *Humanoids*, 2009.

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

[33] A. Saxena, L. Wong, and A. Ng. Learning grasp strategies with partial shape information. *Association for the Advancement of Artificial Intelligence*, 2008.

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

[35] S. Thrun and B. Wegbreit. Shape from symmetry. In *IEEE International Conference on Computer Vision*, volume 2, pages 1824–1831, 2005.

[36] J. Xu, M. Wang, H. Wang, and Z. Li. Force analysis of whole hand grasp by multifingered robotic hand. In *IEEE International Conference on Robotics and Automation*, 2007.

[37] L. E. Zhang, M. Ciocarlie, and K. Hsiao. Grasp evaluation with graspable feature matching. In *RSS Workshop on Mobile Manipulation: Learning to Manipulate*, 2011.