# Knowledge-based reasoning from human grasp demonstrations for robot grasp synthesis

Diego R. Faria [a], Pedro Trindade [a,b], Jorge Lobo [a,*], Jorge Dias [a,c]

[a] Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra - Polo II, 3030-290 Coimbra, Portugal
[b] Citard Services Ltd., Cyprus
[c] Robotics Institute, Khalifa University, Abu Dhabi, United Arab Emirates

A B S T R A C T

Humans excel when dealing with everyday manipulation tasks, being able to learn new skills, and to adapt to different complex environments. This results from a lifelong learning, and also observation of other skilled humans. To obtain similar dexterity with robotic hands, cognitive capacity is needed to deal with uncertainty. By extracting relevant multi-sensor information from the environment (objects), knowledge from previous grasping tasks can be generalized to be applied within different contexts. Based on this strategy, we show in this paper that learning from human experiences is a way to accomplish our goal of robot grasp synthesis for unknown objects. In this article we address an artificial system that relies on knowledge from previous human object grasping demonstrations. A learning process is adopted to quantify probabilistic distributions and uncertainty. These distributions are combined with preliminary knowledge towards inference of proper grasps given a point cloud of an unknown object. In this article, we designed a method that comprises a twofold process: object decomposition and grasp synthesis. The decomposition of objects into primitives is used, across which similarities between past observations and new unknown objects can be made. The grasps are associated with the defined object primitives, so that feasible object regions for grasping can be determined. The hand pose relative to the object is computed for the pre-grasp and the selected grasp. We have validated our approach on a real robotic platform—a dexterous robotic hand. Results show that the segmentation of the object into primitives allows to identify the most suitable regions for grasping based on previous learning. The proposed approach provides suitable grasps, better than more time consuming analytical and geometrical approaches, contributing for autonomous grasping.

*Keywords:*
Robot grasp synthesis, Human grasp demontrations, Object shape representation, Probabilistic inference

## 1. Introduction

As humans stand out in manipulation tasks – a basic skill for our survival and a key feature in our manmade world of artefacts and devices – human manipulation actions and choices can be observed, learned and used to allow a robot with cognitive skills to interact and manipulate objects in our environment. In this work we address an artificial system for grasp synthesis useful for autonomous grasping. The proposed approach relies on knowledge from previous human grasping of pre-defined objects recorded from both human hand and objects' points of view. A learning process is adopted to quantify the probability distributions and the uncertainty over the human grasp experiences. These distributions are combined with preliminary knowledge of grasping choice for specific object shapes towards inference of proper grasping given an object point cloud coming from the sensor observations (RGB-D camera). To accomplish our goal of generating grasp hypothesis given an unknown object, our system is designed in a twofold process: object decomposition and grasp synthesis. This way, after the object decomposition we can find suitable regions for grasping, as well as the candidate grasps for this object. Fig. 1 depicts an overview of our proposed approach.

In a first stage, the decomposition of objects into geometrical shapes is used, across which similarities between past observations and new unknown objects can be made. Afterwards in a second step, the grasp synthesis process generates a list of candidate grasps for the given object. To grasp an object using an articulated hand, we can face the problem of a huge number of grasp configurations possibilities due to the number of degrees of freedom of

* Corresponding author.
*E-mail addresses:* fariadiego@gmail.com; diego@isr.uc.pt (D.R. Faria),

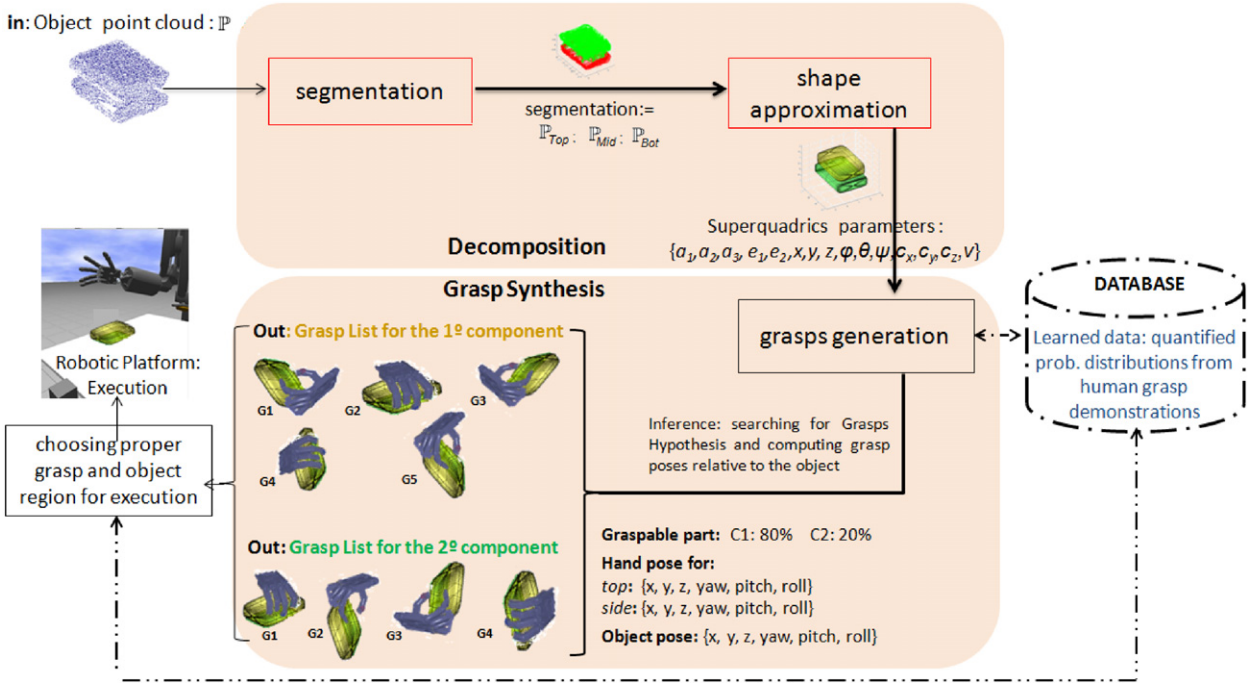



**Fig. 1.** Overview of our proposed approach, broken down into object decomposition and grasp synthesis modules.

the hand. A way to simplify this problem is to adopt a grasp taxonomy that encloses hand configurations that are often used by humans to grasp specific object shapes in different contexts. This way, an object shape can be associated with grasp types by observing humans performing some tasks. We are limiting the huge amount of candidate grasps into a smaller set of probable grasps, since we are simplifying the object shape into primitives and using a subset from a taxonomy of human grasp types. We are using a pre-defined grasp list comprising of 33 grasp types [1] that can also be used for unknown objects. Given a set of hand configurations for a specific object shape, the hand poses relative to the object is computed for the selected grasp from the hypothesis list generated for the object. Using this strategy, we have validated our approach on a real robotic platform — a dexterous robotic hand [2].

This work contributes with a solution that integrates different techniques into a single framework to achieve a full system of grasp synthesis, starting from the object model acquisition up to the grasp execution, that at each step minimizes processing time so as to have an acceptable robot performance time. The artificial system relies on the fact that humans use previous partial knowledge and naturally grasp objects in a stable and proper way in different circumstances, such as objects that differ in size, geometry and orientation. We introduce an ad-hoc solution for object segmentation and subsequent approximation of each of the object components with a geometrical primitive. This breakdown allows matching these primitives with known object components for which sets of viable grasps are known. We introduce a learning stage where the system, given the sensors readings, automatically labels the contact points and object graspable regions using an occupancy grid-based method. The learning stage indexes the graspable regions with a set of candidate grasps. A probabilistic framework is presented in such a way that previous observed knowledge is used for decision making on how to grasp an unknown object.

This article is organized as follows: Section 2 describes related work; Section 3 describes the object segmentation and shape approximation given an object point cloud to define object components for grasping. Section 4 presents how we use the human demonstrations to learn relevant information for reasoning, i.e., inferring how to generate proper grasps for a specific object, and estimating the proper region on the object for grasping. Section 5 presents the grasp synthesis architecture that encloses the steps mentioned in previous sections and the generation of grasp pose relative to the object. Section 6 presents the experimental results, including simulations and also an example in a real application using a dexterous robotic hand. Section 7 draws some conclusions and presents future work directions.

## 2. Related work

Grasp synthesis can be achieved by analytical or empirical approaches. Analytical approaches select the finger positions and hand configurations with kinematical and dynamical formulations. Thus, they generally optimize an objective function such as the grasp stability or the task requirements. On the other hand, empirical (knowledge-based or data-driven) approaches use a learning strategy to choose grasps that depend on the task and on object shape. They rely on sampling candidate grasps for an object and ranking them according to a determined measure or metric [3]. An important factor that needs to be considered in the development of robotic grasping is ensuring stability during the object grasping. There are many approaches for robotic grasping that try to solve this problem, but constraints are encountered. One is finding suitable stable grasps where the task requirements are involved, making the process more complex. To model a robotic grasping, generally, a set of constraints has to be satisfied. First, the robotic hand kinematics and capabilities must be considered. Second, the object features must be taken into account. Finally, the constraints of the task requirements must be analyzed.

When dealing with Learning-by-Demonstration strategies, we can find different works in the literature where the robot observes a human performing a task and afterwards it is able to perform the task by itself. Examples of kinaesthetics demonstrations for learning and generalization, and situated multimodal interaction to teach a robot are demonstrated by [4,5], respectively. Mirror neurons modelling is also a possibility when observing an action in grasping context as demonstrated by [6]. One of the difficulties arising in human based learning is how to measure human performance. Many researchers use data gloves for mapping of human hand to artificial hand workspace and learn the different

joint angles [7,8], or the corresponding task wrench space [9] in order to perform a grasp. Stereo vision is often used to track the demonstrators hand performing a grasp [10] or try to recognize the hand shape from a database of grasp images [11]. Another alternative is to combine different sensors, as presented in [12], which uses a data glove for haptic exploration of objects and vision to track the wrist pose to acquire the object model to be used by a robotic platform. The research presented by [13] shows an example of kinaesthetic teach-in task where the robot learn skills from the demonstrations. The approach uses a mixture of motor primitives that allows the generalization of the robot movements to a wider range of situations. The work presented by [14] automatically extracts action primitives and the corresponding grammar from continuous movements of several human demonstrations of grasping tasks. The approach considers that all the actions can be described by a set of elementary build blocks and there are a set of rules (grammar) that define how these actions primitives can be combined. The action primitives are represented by parametric Hidden Markov Models. The research developed in [15] relies on a probabilistic approach for learning and imitation. Their work addresses the learning of affordances encoding the relationship between actions, objects and effects by the interaction of a robot with the environment. The visual perception plays an important role in their work.

Grasping strategies based on object observation analyse its properties and associate them with different grasps. Some approaches associate grasp parameters or hand shapes to object geometric features in order to find good grasps in terms of stability [16,17]. Other techniques learn to identify grasping regions in an object image [18–20]. In [21] objects are modelled on a set of basic geometrical primitives and rules to generate a set of grasp in starting positions and pre-grasp shapes are defined. By using hand pre-shapes, this method can limit the huge number of possible hand configurations for grasp planning. The planner requires a manually constructed primitive decomposition of the object, so that in [22] they removed the need for a manual decomposition and introduced a multi-level superquadrics representation. Different works use methods such object geometry or object segmentation which can facilitate the grasp hypothesis generation [23,24]. In [23] an iterative segmentation algorithm for grasping non-convex objects is proposed. First, the inertial axes of the whole object are computed for grasp generation. When the tentative to obtain valid grasps fails, the object decomposition process starts. At each iteration of the decomposition step, two components are obtained towards reaching feasible grasps. The process is repeated until a grasp is found or the decomposition terminates. The constituting parts of an object shape influence the choice of an object graspable part, independently of their orientation. The relative size of the object component is very important to select the graspable part [25,26]. Before achieving a good strategy for grasping based on object observations, the object detection has to be well performed. Many works on grasping are based on vision systems, such as [27], and also including RGB-D sensors [28].

The research carried out in [29] presents a model for high level planning of in-hand manipulation tasks using a dexterous robotic hand. A sequence grasp transitions are autonomously generated from the initial grasp. The choice of the successive grasps is modelled as a Markov Decision Process, using the probability of success of the transitions between canonical grasps to generate a policy modelled as a Bayesian network. The strategy can be applied for tasks where grasp transitions are needed. The work developed in [30] draws on the synergy idea of the fingers joint movements being adaptive during the grasping. The authors presented an algorithm that receives as input the joint trajectories that can adapt to the real contact pressure of each finger in order to guarantee stability, avoiding the slippage or contact breaking during the manipulation. In [31] a grasp synthesis solution using a compliance model to synthesize grasps is proposed. The solution is inspired on soft-synergies using a kinestatic formulation, and it is considered as a constrained optimization problem introducing compliance to address the constraints of contact reachability, object restraint and force. Other recent work on the analysis of natural and robotic hand presents the postural synergies (trajectory in configuration space assuming a smaller dimension than the kinematic) based on the physical characteristics of the hand is presented in [32].

Some of the works previously mentioned [21–26] use object decomposition into parts to define a small search space that is likely to contain many grasps. In our work we use object decomposition, but unlike the previous approaches, we also use knowledge from human demonstrations combined with features from the object model, and hence, suitable grasp regions are associated with the corresponding grasp configurations as probability distributions for the components. When new unknown objects are observed, an inference is made by decomposing the new object and matching with the previous knowledge. In particular, the works presented in [21, 23,25], that model objects with superquadrics or use a decomposition strategy, only present simulation results. Our system can generate grasp hypothesis not only in simulation, but on-the fly for a real robotics platform given an unknown object. Processing time is a key concern to have an acceptable performance. To find object meaningful parts, we avoid time consuming algorithms based on mesh segmentation such as used in [25,33], and rely on an ad-hoc solution for object component segmentation, that combined with subsequent matching with learned grasps, allows finding suitable graspable regions. The proposed framework is based on the observed human choices, assigning a limited number of ranked grasp configurations (candidate grasps) for each shape primitive. Unlike other works that manually label the object regions after observing the human grasping [25,26], we rely on an automated process. During the demonstrations, sensed contact point locations enable computation of grasps types and of the object graspable regions. Our framework follows a probabilistic approach to build a full system of grasp synthesis, taking decisions based on previous observed data to generalize to other contexts.

## 3. Object decomposition: segmentation and shape modelling

Humans usually identify object parts in order to search for a suitable region for grasping. The Recognition By Components theory (RBC) [34] reveals that humans are able to identify objects by segmenting them into geometric shapes. If we see an unfamiliar object, despite its unfamiliarity, we are able to identify this object by segmenting it into parts at regions of deep concavity, looking for known or familiar shapes. Based on this study, we are decomposing the object point cloud to find meaningful parts for grasping. In a first step, we present the object segmentation process, and later, the shape approximation into geometrical primitives. This process will be used in the learning and grasp synthesis.

### 3.1. Object segmentation

We proposed an ad-hoc solution for a fast segmentation of everyday objects enabling a real time performance for a robotic platform. To validate the object decomposition method, we have used a set with a few everyday objects acquired from different modalities (RGB-D camera, laser-scanner sensor and shape acquisition from in-hand manipulation [35,36]) to test the method. The set of everyday objects used in this section is presented in Figs. 2 and 8: bottle, ladle, mug, Rubik cube, sponge, spray-bottle, wii-mote, wii-nunchuck and a wooden cat.

The following subsections will present two different solutions proposed for object segmentation, and their evaluation taking


**Fig. 2.** Examples of everyday objects with different shapes and sizes that were used to test the decomposition approach.

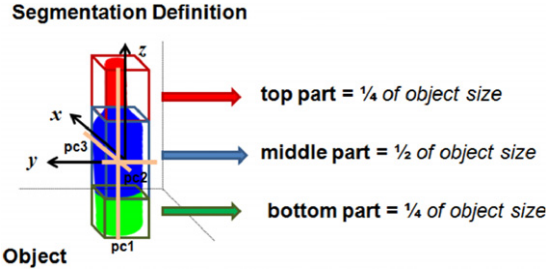



**Fig. 3.** Object Segmentation Definition. The object is segmented into three parts: top, middle and bottom if the object is not so small. The segmentation takes into consideration the major axis (pc1: principal component), i.e., the axis with the bigger length.

into account the processing time and a qualitative analysis to point out the most feasible solution for the next stage of shape approximation. The first solution is based on the object major axis, and the second one is based on clustering by Gaussian mixture models.

### *3.1.1. Segmentation based on major axis analysis*

For the segmentation, we are assuming that all everyday objects are composed by Top, Middle and Bottom parts if the object size satisfies a defined threshold of size based on the assumption that an object cannot be too small that it cannot be grasped properly for a specific taxonomy, and not too big that the robotic hand cannot grab it.

The threshold for segmentation is based on the assumption that humans have knowledge about what is a small, medium or big object for grasping taking into consideration the hand size and previous knowledge learned from lifelong experience. Thus, we defined 7 cm as a starting point for segmentation. If an object is smaller than that (e.g. a Rubik cube), the segments would be very small and to apply a power grasp on one of the components would be a hard task, making it difficult for a robotic hand to properly grasp the object region. Fig. 3 shows an example of the segmentation strategy. We use the object-centred representation, and the object frame of reference is found based on its centre (origin). The axes are defined as $\{x, y, z\}$: right-hand rule ($x$: index finger pointing to front; $y$: middle, pointing to the left; and $z$ thumb, pointing up).

The segmentation process is based on the idea of methods that analyse the major axis, such as the known method in the state of the art, Principal Coordinate Analysis (PCoA), which is a related statistical technique often used in information visualization for exploring similarities or dissimilarities of the data. The idea here is simple; we arrange the data by the major axis based on distance measures. More specifically, given a set of the 3D points $\mathbb{P}$ that form an object, where a 3D point follows the notation $\boldsymbol{P}_i = \boldsymbol{P}(x, y, z, r, g, b) \in \mathbb{P}$, where $\{x, y, z\}$ are Euclidean space coordinates and $\{r, g, b\}$ the RGB colour components, we search for the axis vector with higher magnitude. By analyzing the points in each axis $\{x, y, z\}$, we can search the points with maximum and minimum coordinate values to compute the object length in the Cartesian space, using the distance between these points. Let $\vec{e}$ be a vector with the points at maximum and minimum coordinate in a specific axis $\{x, y, z\}$, which may take the following forms: $\mathbf{e}_x = \{x_{\min}, x_{\max}\}$, $\mathbf{e}_y = \{y_{\min}, y_{\max}\}$, $\mathbf{e}_z = \{z_{\min}, z_{\max}\}$. Then the higher magnitude is computed using the Euclidean distance as follows:

$$\|\mathbf{e}_a\| = |\mathbf{e}_{\max} - \mathbf{e}_{\min}|, \tag{1}$$

where $\mathbf{e}_{\max}$ is the first element of $\mathbf{e}$, representing the point at the maximum coordinate in a specific axis and $\mathbf{e}_{\min}$ is the second element of $\mathbf{e}$, representing the point at the minimum coordinate in the same axis. Then we search for the major axis $a$ as follows:

$$a = \begin{cases} \{x\}, & \|\mathbf{e}_x\| > \|\mathbf{e}_y\| > \|\mathbf{e}_z\| \\ \{y\}, & \|\mathbf{e}_y\| > \|\mathbf{e}_x\| > \|\mathbf{e}_z\| \\ \{z\}, & \|\mathbf{e}_z\| > \|\mathbf{e}_x\| > \|\mathbf{e}_y\| \,. \end{cases} \tag{2}$$

Afterwards, the segmentation is applied based on the object axis vector with higher magnitude. Let $\mathscr{B}$ be a specific boundary (top or middle or bottom region) of the object point cloud. Each 3D point $\boldsymbol{P}_i$ will belong to a specific region (i.e., $\mathscr{R}_t$: top; $\mathscr{R}_m$: middle; $\mathscr{R}_b$: bottom), if this point is inside of that region boundary. The boundary verification is achieved by the following steps:

$$\boldsymbol{P}_i \in \mathscr{R}_t : (\boldsymbol{P}_i^a \geq a_{\max} - \mathscr{B}_t^a), \tag{3}$$

$$\boldsymbol{p}_i \in \mathscr{R}_b : (\boldsymbol{P}_i^a \leq a_{\min} + \mathscr{B}_b^a), \tag{4}$$

$$\boldsymbol{P}_i \in \mathscr{R}_m : (a_{\min} + \mathscr{B}_b^a) < \boldsymbol{P}_i^a < (a_{\max} - \mathscr{B}_t^a), \tag{5}$$

where $\boldsymbol{P}_i$ is a point that belongs to the object point cloud $\mathbb{P}$, $i = \{1, \ldots, n\}$; $\boldsymbol{P}_i^a$ is the point's coordinate in the major axis $a$; $a_{\max}$ and $a_{\min}$ are the points in the major axis at the maximum and minimum coordinate, respectively; $\mathscr{B}_t^a$ is the boundary for $\mathscr{R}_t$ in $a$; $\mathscr{B}_b^a$ is the boundary for $\mathscr{R}_b$ in $a$; $\mathscr{B}_m^a$ is the boundary for $\mathscr{R}_m$ in $a$. The

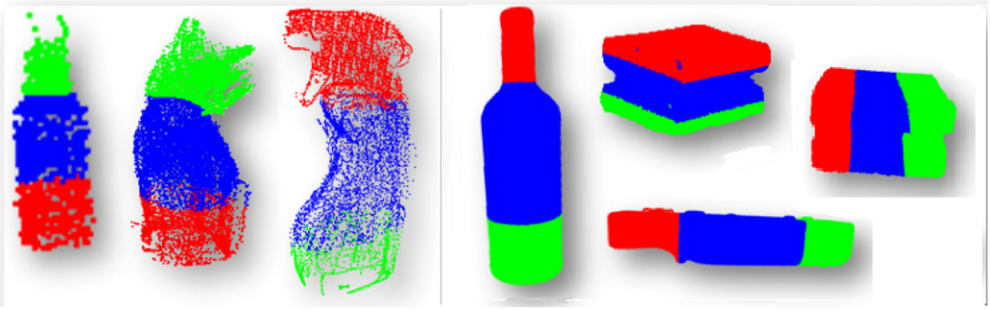


**Fig. 4.** Segmentation results for everyday objects based on the major axis. The models acquired by in-hand exploration are shown on the left, and on right side the models acquired using a laser scanner. The right side shows an example (sponge) where the major and the second axis are close in size, so that we have segmented both to compare the results.

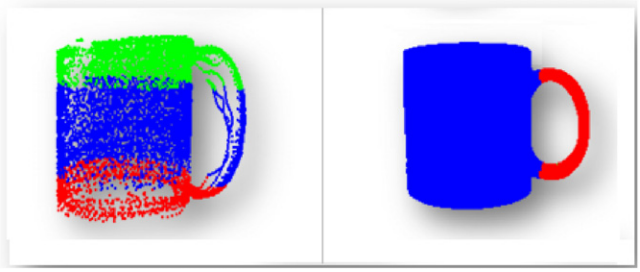


**Fig. 5.** Mug segmentation: (left) a mug acquired by in-hand exploration segmented into 3 subcomponents; (centre and right) a mug acquired using a laser scanner segmented into 2 subcomponents, since the dimensions are very similar in two directions either the vertical or the horizontal axis can end up being selected.

region boundaries were chosen taking into account the expected size range of the objects, as follows:

$$\mathscr{B}_t = \begin{cases} \dfrac{|a_{\max} - a_{\min}|}{4}, & |a_{\max} - a_{\min}| \geq 7 \text{ cm} \\ \dfrac{|a_{\max} - a_{\min}|}{2}, & 5 \text{ cm} \leq |a_{\max} - a_{\min}| < 7 \text{ cm} \\ |a_{\max} - a_{\min}|, & |a_{\max} - a_{\min}| < 5 \text{ cm} \end{cases} \tag{6}$$

$$\mathscr{B}_m = \frac{|a_{\max} - a_{\min}|}{2}, \tag{7}$$

$$\mathscr{B}_b = \mathscr{B}_t, \tag{8}$$

where the boundaries $\mathscr{B}_m$ and $\mathscr{B}_b$ are used only if the object size is bigger then a determined threshold. Algorithm 1 shows a summary version for object segmentation.

Fig. 4 presents the results achieved for some everyday objects to validate the segmentation method. The object point clouds on the left were achieved by in-hand exploration [36,35] and on the right by laser scanner.

Fig. 5 presents two different cases of segmentation and different number of object components. For the mug object, the two scanning methods had two distinct results. This will happen for objects that do not have a clear major axis, and have similar dimensions along two axes. The segmentation method for the mug acquired by the laser scanner selected the $y$ direction as the major axis.

The segmentation of everyday objects into three components can describe different candidate regions for grasping. Searching for optimal or plausible contact points on the entire geometry of the object (e.g. searching on the mesh) is time consuming, examples of these case are given when GRASPIT! simulator [37] is used to generate the grasps, and depending on the size of the object point cloud the grasp hypothesis can take minutes. For real applications on a robotic hand, the mesh would first have to be computed, followed by additional computations of stable

**Algorithm 1:** Object components segmentation algorithm based on major axis analysis

1 Input: Object Point Cloud $= \mathbb{P}$;

2 $\forall \boldsymbol{P}(x, y, z, r, g, b) = \boldsymbol{P}_i \in \mathbb{P}$, search for the points with maximum and minimum coordinates values in each axis $\{x, y, z\}$, and build the vectors: $\mathbf{e}_x = \{x_{max}, x_{min}\}$, $\mathbf{e}_y = \{y_{max}, y_{min}\}$, $\mathbf{e}_z = \{z_{max}, z_{min}\}$ ;

3 $\forall \mathbf{e}$ compute their magnitudes ($\|\mathbf{e}_x\|$, $\|\mathbf{e}_y\|$, $\|\mathbf{e}_z\|$);

4 Compare the magnitudes of the vectors $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$ and keep the biggest one considering its reference $x$ or $y$ or $z$ as the major axis $a$ ;

5 Verify the object size (in distance, e.g. cm) in the major axis, and search the regions Top: $\mathscr{R}_t$; Middle: $\mathscr{R}_m$ and Bottom: $\mathscr{R}_b$, by computing the boundaries $\mathscr{B}_t$, $\mathscr{B}_m$, $\mathscr{B}_b$ as demonstrated in Eqs. (6)–(8).

6 Segment the object by labelling the points in red if $\boldsymbol{P}_i \in \mathscr{R}_t$, or in blue if $\boldsymbol{P}_i \in \mathscr{R}_m$ or in green if $\boldsymbol{P}_i \in \mathscr{R}_b$ by computing Eqs. (3)–(5).

7 Outputs: Object Segments (top, middle and bottom) $= \mathbb{P}_{top}$, $\mathbb{P}_{mid}$, $\mathbb{P}_{bot}$

grasping regions. In our case, by segmenting the objects into three components, we can approximate each segment by a geometrical primitive (e.g., quadrics) allowing an association with previously observed candidate grasps for each geometrical primitive. Cases of non-elongated or small objects, for instance, a cube and a ball are initially considered as a single segment due to its size, and afterwards they are approximated to a single shape primitive as a cube and sphere. In these cases of a single segment, the points around the centroid of the object will be indicated as graspable region.

We have an implicit assumption that the grasps are adaptive by using synergies of the fingers [38]. This approach does not require the association of an exact geometry of the object with exact grasp geometry, since the given approximated grasp type is dynamically adjusted using synergies when grasping. Our strategy of object decomposition into components does not require a complex segmentation avoiding high computational processing time, being sufficient and with high possibility of grasping success for everyday objects, which justifies our segmentation method.

To evaluate the effectiveness of the proposed segmentation based on the major axis, we have compared with another method based on clustering techniques by means of Gaussian Mixture Models (GMM) introduced in our previous works [36,39] and presented in the next subsection.

*3.1.2. Segmentation based on Gaussian mixture models*

In this subsection, we are addressing a segmentation method by means of Gaussian Mixture Models given the object point cloud. It is another alternative that allows for the search for segments on the object which are candidate regions for grasping. The estimation of the parameters (e.g., mean, covariance matrix and weight) of each individual Gaussian density function (cluster) is accomplished by the Expectation Maximization (EM) algorithm, also known as *EM clustering*, which is an iterative method that attempts to find the estimator with the maximum likelihood of a parameter. A global parameter that needs to be set is the maximum number of clusters $k_{\max}$. An optimal $k_{\max}$ can be estimated by MDL (Minimum Description Length) penalty function [40] on the input data. In our case we pre-set $k_{\max} = 3$, since we have observed that it is sufficient for hand-held everyday objects.

Let a set of points (i.e., object point cloud represented by a matrix of points) be $\mathbb{P} \in \mathbb{R}^3$, generated independent and identically distributed (i.i.d.) by a mixture of $k$ Gaussians, and $\wp_i \subset \mathbb{P}$ representing a subset of the point cloud, that is, a specific cluster $j = \{1, \ldots, k\}$. Each set or subset of 3D points encloses many 3D points, $\boldsymbol{P}(x, y, z) = \boldsymbol{P}_i \in \wp_i \subset \mathbb{P}$. The entire set of parameters denoted as $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$, where $\mu_j$ represents the mean of a cluster $\wp_i$, $\Sigma_j$ represents the covariance matrix and $w_j$ represents the weight of the same cluster, specifies how likely each Gaussian is selected. The derivation of EM algorithm to estimate the set of GMM parameters $\theta$, for $\mathbb{P}$ (input) and any $\mu_j$, $\Sigma_j$, is denoted as Gaussian according to the following expression:

$$\phi(\wp_i|\mu_j, \Sigma_j) \triangleq \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(\wp_i - \mu_j)^T \Sigma_j^{-1}(\wp_i - \mu_j)\right). \quad (9)$$

The *pdf* for the combination of the $k$ models to search for the most likely combination $\theta$ of models to explain the observed data is achieved by (10). This means a learning of mixture models, so that we are searching for the combination of the proper clusters that better describes the input data $\mathbb{P}$, achieving the subset of points $\wp_i$, representing the proper cluster $j$, where $j = \{1, \ldots, k\}$.

$$P(\wp_i|\theta) = \sum_{j=1}^{k} w_j \phi(\wp_i|\mu_j, \Sigma_j), \quad (10)$$

where $w_j > 0$, $\sum_k^j w_j = 1$ and $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$.

To summarize the EM algorithm that estimates the GMM parameters, we apply then the following steps, first compute the initial log-likelihood, that is used later to check the convergence of the EM algorithm:

$$\ell^{(0)} = \frac{1}{n}\sum_{1}^{n} \log\left(w_j^{(0)} \phi(\wp_i|\mu_j^{(0)}, \Sigma_j^{(0)})\right), \quad (11)$$

where $n$ is the amount of samples contained in $\wp_i$; the initial estimates $w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = \{1, \ldots, k\}$ can be randomly chosen. During the initialization, we can take some $k$ of the object point cloud $\mathbb{P}$ as the first estimate of the cluster mean, setting the first estimate of the covariances to be the identity matrices, and the first guess at the weights $w_i = \cdots = w_k = 1/k$, which is common when using this algorithm. A better alternative commonly used, and also adopted in our work, is adopting the $K$-means algorithm to provide a good initialization for the EM. The **E** (Expectation) step is achieved by (12). Let $\gamma_{ij}^m$ be the estimate at the $m$th iteration of the probability that the $i$th sample was generated by the $j$th Gaussian component (cluster), as demonstrated as follows:

$$\gamma_{ij}^m = \frac{w_j^{(m)} \phi(\wp_i|\mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{j=1}^{k} w_j^{(m)} \phi(\wp_i|\mu_j^{(m)}, \Sigma_j^{(m)})}, \quad i = \{1, \ldots, n\}. \quad (12)$$

**Algorithm 2:** EM algorithm for estimating GMM parameters

1 Inputs: Object point cloud $\mathbb{P}$

2 **Initialization**: Choose the initial estimates $w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = \{1, \ldots, k\}$, and compute the initial log-likelihood as demonstrated in Eq. (11).

3 **while** $|\ell^{(m+1)} - \ell^{(m)}| > \delta$ *(pre-set threshold)* **do**

4 **E step**: For $j = 1, \ldots, k$, compute $\gamma_{ij}^m$ and $s_j^{(m)}$ as exemplified in Eqs. (12)–(13)

5 **M step**: For $j = 1, \ldots, k$, compute the new estimates: $w_j^{(m+1)}, \mu_j^{(m+1)}$ and $\Sigma_j^{(m+1)}$ as demonstrated in Eqs. (14)–(16)

6 **Convergence step**: compute the new log-likelihood $\ell^{(m+1)}$ as shown in Eq. (17)

7 Outputs: $\theta = \{(w_j, \mu_j, \Sigma_j)\}_j^k$

To facilitate the representation of the next formulas, we use a notational simplification, denoting the total membership weight of the $j$th cluster as $s_j^{(m)}$ as follows:

$$s_j^{(m)} = \sum_{i=0}^{n} \gamma_{ij}^{(m)}. \quad (13)$$

Consequently, the **M** (maximization) step is given by:

$$w_j^{(m+1)} = \frac{s_j^{(m)}}{n}, \quad (14)$$

$$\mu_j^{(m+1)} = \frac{1}{s_j^{(m)}} \sum_{i=0}^{n} \gamma_{ij}^{(m)} \wp_i, \quad (15)$$

$$\Sigma_j^{(m+1)} = \frac{1}{s_j^{(m)}} \sum_{i=0}^{n} \gamma_{ij}^{(m)} \left(\wp_i - \mu_j^{(m+1)}\right)\left(\wp_i - \mu_j^{(m+1)}\right)^T, \quad (16)$$

where the maximization step is computed for all clusters $j = \{1, \ldots, k\}$. Afterwards, the new log-likelihood is computed to verify the convergence of the algorithm $|\ell^{(m+1)} - \ell^{(m)}| > \delta$ (pre-set threshold) as follows:

$$\ell^{(m+1)} = \frac{1}{n}\sum_{1}^{n} \log\left(w_j^{(m+1)} \phi(\wp_i|\mu_j^{(m+1)}, \Sigma_j^{(m+1)})\right). \quad (17)$$

All steps mentioned above are summarized in Algorithm 2. More details about the theory and use of the EM algorithm and the GMM learning can be found in [41].

Afterwards, each cluster generated by the EM clustering is represented as a segmented region of the object that can be used as a candidate region for grasping. Examples of the segmentation using the GMM method are shown in Fig. 6.

The success of the GMM method depends on how the clusters are generated, taking also into consideration the amount of clusters. Sets with a larger number of points have a significant impact on the algorithm's processing time, due to the iteration steps to estimate the parameters of each cluster. When $k_{\max}$ is defined to be more than three clusters, then the results for everyday objects are not so satisfactory, because sometimes an object with too many segments does not present reasonable candidate regions for grasping. Some examples of segmentation with little success (based on a qualitative analysis) are presented in Fig. 7.

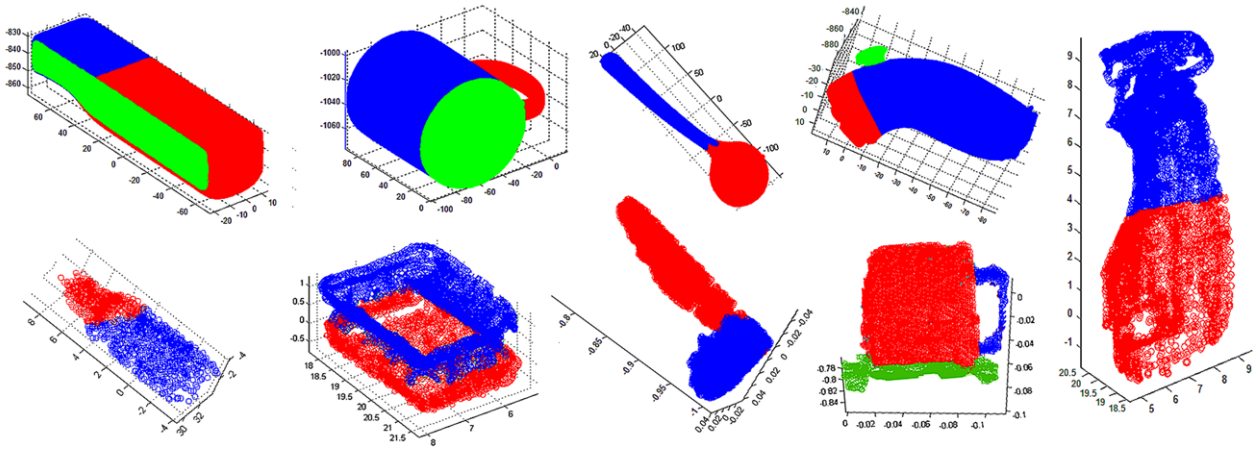


**Fig. 6.** Everyday objects (wii-mote, mug, sponge, bottle, ladle, Nintendo nunchuck and spray bottle) segmentation using GMM clustering. These objects were acquired by different sensor modalities to test the segmentation. Top row: laser scanner; bottom row and last image (at right): in-hand exploration (bottle, sponge and spray bottle); RGB-D device (ladle and mug).

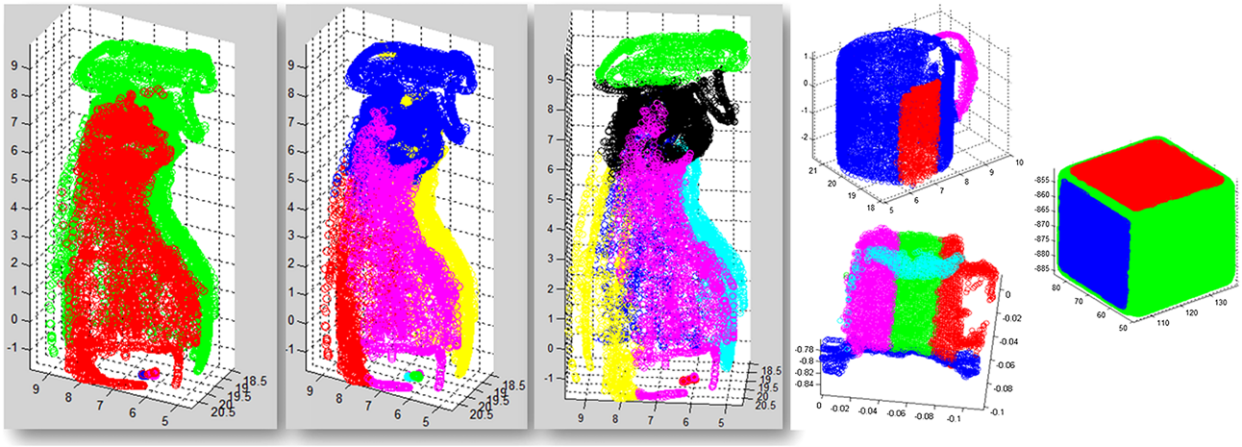


**Fig. 7.** Examples of segmentation of everyday objects with little success using GMM clustering. Some segments cannot be considered as a good candidate region for grasping (based on a qualitative analysis). Some of the segmented regions are not suitable for subsequent approximation by a geometrical primitive.

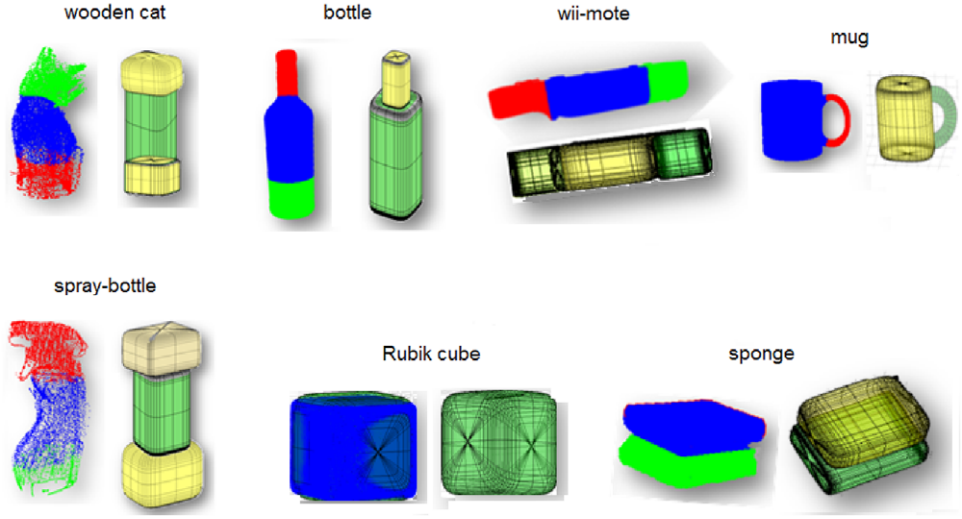



**Fig. 8.** Shape approximation: superquadrics models obtained for the segmented everyday objects.

### *3.1.3. Discussion of the segmentation methods*

Using the segmentation solution based on the major axis we could achieve satisfactory results for everyday objects. The algorithm is fast (10 ms on a standard PC) even for an object point cloud with a large number of points (200 000), unlike other methods that are more time consuming when the point cloud is larger. The more elaborated methods [33,25] using meshes based on Gaussian curvatures try to find meaningful and functional parts of the object covering many steps, making it computationally expensive (taking almost one minute with similar hardware). For robot grasping applications that require real time processing our simple method based on the major axis has a clear advantage.

Although the method uses a fixed number of subcomponents (top, middle and bottom), we could observe that, even in degenerate cases, grasp types could be associated with these regions, thereby allowing the robotic hand to grasp one of these regions. This algorithm proved to be a fast ad-hoc working solution for everyday objects. Some tweaking to better define the boundaries for the segmentation can help dealing with more complex objects, with more concave or convex regions, or even with many handles, but other methods indicated above should be used for those cases. The main problem with shape driven methods that search for object functional parts is that the larger the object point cloud, the longer the processing time, unlike of our approach that will always have a similar processing time, i.e., independent of the point cloud size.

### 3.2. Object shape modelling

Having segmented the object, we now want to model each segment as a basic geometrical primitive. In this work we use superquadrics [42], a technique that models a rich variety of shapes, and that facilitates computing parameters that enclose important cues, such as scale and orientation. Superquadrics has been used for 3D object modelling [43] and for segmentation of point cloud [44], in robotics (novelty detection) [45] and successfully in other works for grasping purposes [21,25,24].

The superquadrics models are expressed by a function $\boldsymbol{f} : \mathbb{R}^3 \rightarrow \mathbb{R}$ as:

$$\boldsymbol{f}(x, y, z) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \frac{y}{a_2}^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}, \tag{18}$$

where $\epsilon_1$ and $\epsilon_2$ are the parameters for shape; $a_1$, $a_2$ and $a_3$ are the scale factors on the $\{x, y, z\}$ axes. This form provides information on the position of a 3D point relative to the superquadric surface. The implicit function $\boldsymbol{f}(x, y, z)$ partitions the space into three regions: the point $\boldsymbol{P}(x, y, z)$ lies on the surface if $\boldsymbol{f}(x, y, z) = 1$, if $f(x, y, z) < 1$ then the point is inside, and outside when $\boldsymbol{f}(x, y, z) > 1$. Even if the five parameters of the model are compact, it allows to deal with a large variety of shapes such as cylinders, spheres, ellipsoids, parallelepipeds and others. The shape parameters can be constrained to have, for example, just convex shapes (when $\epsilon_1 < 2$ and $\epsilon_2 < 2$).

The recovery of the superquadrics from a point cloud is represented in a global coordinate system. Thus, we have another 6 parameters to express the rotation (Euler angles $(\phi, \theta, \psi)$) and translation $((p_x, p_y, p_z))$. The function can also be expressed as $\boldsymbol{f}(x, y, z, \Lambda)$, where the set of the 11 parameters can be represented as $\Lambda = \{a_1, a_2, a_3, \epsilon_1, \epsilon_2, p_x, p_y, p_z, \phi, \theta, \psi\}$, representing three parameters for scale in each axis, two parameters for shape variation; and six parameters representing the translation and rotation in each axis, respectively.

After the segmentation process, the set points of each object component will be approximated by a superquadric shape primitive. To estimate the parameters of the superquadric model, the gradient least-square minimization of an error-of-fit function based on Levenberg–Marquardt method [46] is used as follows:

$$\min_{\Lambda} \sum_{i=1}^{n} \left( \sqrt{a_1 a_2 a_3} \left( \boldsymbol{f}^{\epsilon_1}(x_i, y_i, z_i; \Lambda) - 1 \right) \right)^2, \tag{19}$$

where $\sqrt{a_1 a_2 a_3}$ are constraints used to find the smallest superquadric based on the scale parameters. The power $\epsilon_1$ makes the error metric independent of the superquadric shape. More details can be found in [46].

The superquadrics fitting initialization can influence the convergence of the method, i.e., the necessary number of iterations to find the best fitting. In this work, the shape parameters of a superquadric model are initialized as an ellipsoid, $\epsilon_1 = \epsilon_2 = 1$, and the object initial pose is based on the centre of gravity computed from the statistical moments. The scale factors are based on the computed eigenvalues of the computed inertia matrix.

For each object component (segmented region), a superquadric model is generated. Fig. 8 shows the superquadrics models generated for the segments of some everyday objects, simplifying the object parts shape into geometrical primitives.

More parameters can be computed to represent a superquadric model, the 11 parameters already explained, and further 4 parameters that can be included in this set of parameters, such as the centroid of the superquadric $\{c_x, c_y, c_z\}$ and its volume $v$.

We have limited the set of geometrical primitives that can compose everyday objects into the following superquadrics models: box and its variation (rounded box), cube, cuboid, cylinder, ellipsoid, sphere, octahedron, spinning-top (squared, rounded and star shape) and variation 1 of the sphere (arch) and variation 2 of the sphere (butterfly shape). The method smoothly varies (continuously) the shape parameters $\epsilon_1$ and $\epsilon_2$ to fit the best shape given the point cloud. These thirteen superquadrics models are sufficient to approximate and describe a big variety of everyday objects.

The superquadrics models can represent an object shape even when the input is a partial volume of the object. For the everyday objects used in this work, we have achieved results that were satisfactory and useful for the subsequent grasping synthesis. The superquadrics modelling for a point cloud acquired from an RGB-D sensor in average takes less than one second up to three seconds if the point cloud is too dense (e.g., when acquired by a laser scanner sensor, which it is not the case here) using a standard computer (e.g., a Laptop with Intel core i3-3500M processor, 2.26 GHz, 4 GB DDR3 Memory). The precise processing time depends on the object point cloud that varies in shape and size.

We are using the superquadrics modelling to approximate an unknown object shape into a familiar geometric primitive, thus simplifying the huge amount of candidate grasps for this familiar shape. This will reduce the grasp space into the learned grasp (hypothesis) for that familiar shape since we are learning grasp types for a set of geometrical primitives. Decomposing the object into parts will give hypothesis of grasping the object in its different segments.

## 4. Learning from human grasp demonstrations

In this section we address the learning from human grasp demonstration that will assist the grasp hypothesis generation. The learning strategy is based on our previous works [36,47].

The learning phase follows a probabilistic approach, and in general finds a model that describes the dependency of one random variable on another one. Let $q_i \in Q, i = \{1, \ldots, N\}$ be possible object regions (quadrics) and $g_k \in G, k = \{1, \ldots, M\}$ be the possible grasp types, then the dependency is defined by a conditional probability distribution $P(Q|G)$ that is the probability density function ($pdf$) of a random variable representing one of the target classes given the random variable representing the input vector (features). In other words, this means that, given the space of possible inputs $Q$ and the possible targets space $G$, an estimate of the class that encloses the input space is given by a classification model resulting $P(G|Q)$.

The human grasp demonstrations result in a dataset $\mathscr{D}$ with $N$ labelled examples coming from the learning phases as presented subsequently in the following subsections. An example of a labelled dataset is given by the possible candidate grasps $G$ associated with each quadric $q_i$ representing an object region. Each grasp type can be found using the hand fingertips 6D data in the

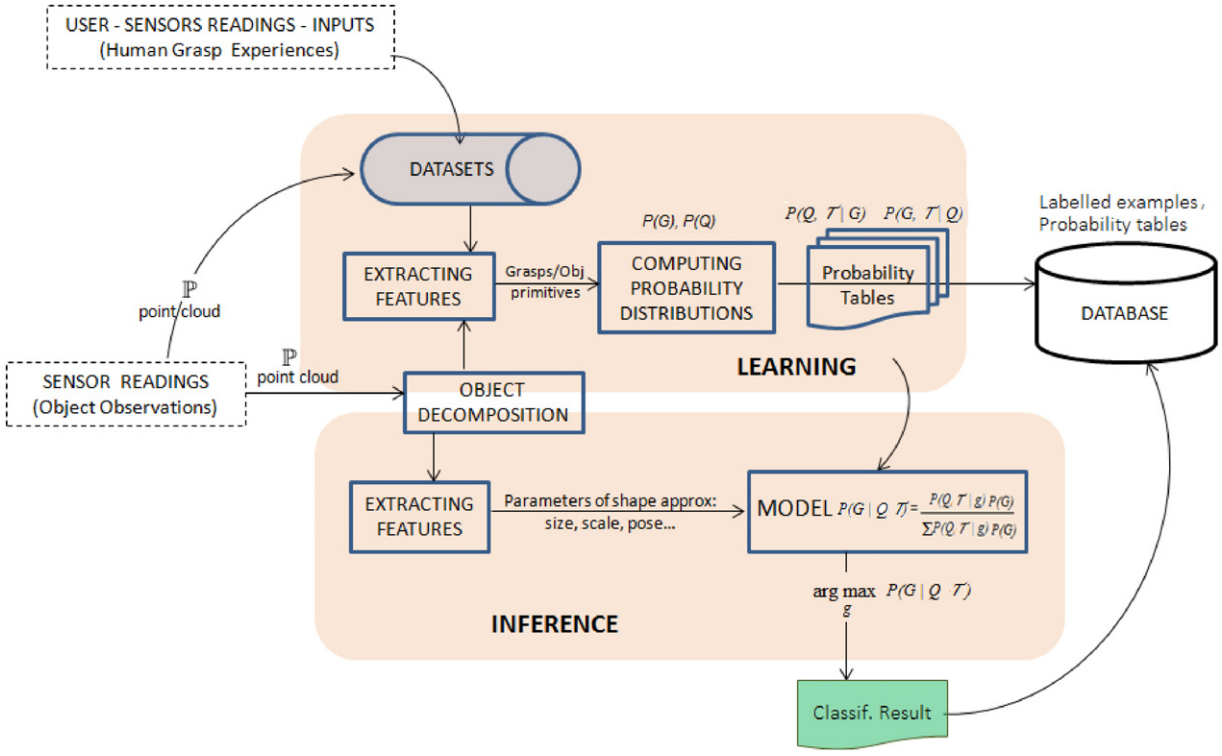



**Fig. 9.** Overview of the learning process assisting the inference to search for candidate grasps for a given object model.

wrist coordinate system, forming a hand configuration. From the observed data, inferences can be made to assist the grasp synthesis to estimate possible grasps given a quadric model, or to search for possible candidate regions on the object to perform a grasp, or even the candidate regions on the object given a task context $\mathscr{T}$. The inference models are based on Bayesian techniques that will be explained afterwards.

Fig. 9 depicts an overview of the learning and inference strategies adopted in this work. In the next subsections, the steps for learning and inference based on human grasping experiences will be described.

### 4.1. Overview of Bayesian inference using the learned data

A general explanation of how we perform inference adopting Bayesian theory is given in this subsection. Later, in the following subsections, we will show the models we have defined that use the likelihoods useful for our grasp synthesis system, taking decisions on which are the most probable grasps and suitable regions for grasping given the object model.

Probabilistic techniques such as Bayesian theory is used to support the decision given the learned likelihood. A strong assumption is that all inputs are mutually independent of each other given the class label (e.g., grasps types associated with object regions under a task context). When adopting a Dynamic Bayesian Network (DBN) for the learned likelihood, the joint probability distribution is represented as a set of random variables. The set of parameters in a DBN encloses the conditional probability distribution of the random variables and the learned probability tables. Then using the Markov condition, each node is stated as independent of its non-descendants given its parents. Fig. 10 presents a general example of an application in a grasp context to better understand the models. It represents the probability of a grasp type happening when some events occur, such as when the artificial system finds a specific object region, represented as a quadric $q_i$, and inside of a task context $\mathscr{T}$. The events (parents) represent the set of parameters $Q$, $\mathscr{T}$ that trigger an effect (node $G$).

The DBN can be used as a classifier that gives the posterior probability distribution of the class node $G$ given the values of other attributes (set of events). The model represented in Fig. 10 can be expressed as the posterior distribution $P(G|Q, \mathscr{T})$ given the observations enclosing the random variables $Q$, $\mathscr{T}$ as follows:

$$P(G|Q, \mathscr{T}) = \frac{P(Q, \mathscr{T}|G)P(G)}{\sum_j P(Q_j, \mathscr{T}|G)P(G_j)}, \tag{20}$$

then, using the MAP estimation $\text{argmax}_{g_k \in G}\ P(G|Q, \mathscr{T})$, we have the classification result. Over time the model presented in (20) can be stated as $P(G|Q, T) = \prod P(Q, \mathscr{T}|G)P(G)$ due to the iteration, i.e., while new observations are coming from the readings. The dynamics in the BN is when the system's state at time $t$ depends only on its immediate predecessor $t-1$. The time instant $t$ of the system evolves over time according to the system dynamics that is specified by the conditional density function $P(G_{t+1}|G_t)$. Initially, in our model, the prior is a uniform distribution, and over time the previous posteriori becomes the next prior, updating this way the model by taking the previous knowledge into consideration.

The general models mentioned in this section were given to exemplify the techniques adopted for inference using the learned data to assist the grasp synthesis. The learned likelihoods used in this work were built based on histogram techniques. The idea is to rely on statistical data to achieve a successful estimate. The learning phases and inferences used in our artificial system are explained in the next subsections.

### 4.2. Experimental setup for data acquisition

In this work, the human demonstrations play an important role in the ability of learning and identification of how to grasp objects. The experimental activities with humans executing grasping tasks are performed in our experimental area with multiple data acquisition devices (Fig. 11) in order to capture how humans perform successful tasks. The data acquired is used to model and

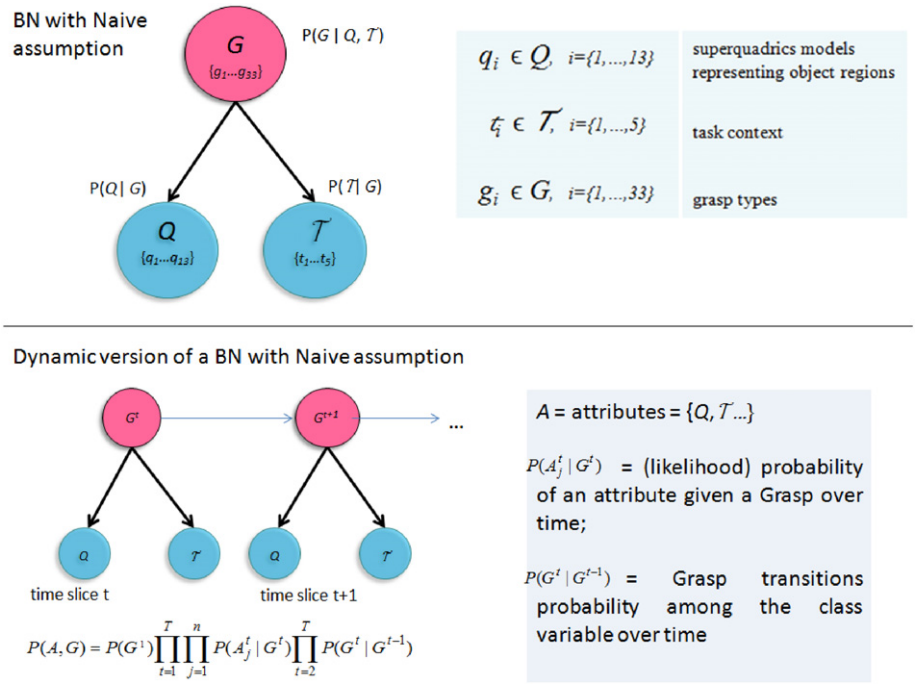



**Fig. 10.** A Bayesian network to represent a general model when a grasp type is estimated given some causes $Q$, $\mathcal{T}$. The image at the top row shows a simple BN with naive assumption (the children nodes are independent from each other). The bottom image is a representation of a dynamic process with two time slice. This model assumes that the attributes in different time slice are independent from each other given the class variable. This assumption implies that the temporal dynamics is captured at class level only. Even though, another network might be modelled with a more complex structure where dependence among all nodes can happen.

**Fig. 11.** Sensors used in our experimental setup: RGB-D camera (MS-Kinect), Polhemus Liberty Magnetic Tracking System, Cyber-Glove and Tekscan Tactile sensors.

extract the relevant aspects of the human demonstration, as well as providing input for the methods presented in this work. For that, we are reading data from two perspectives: the human hand and the object. In the first case, finger 6D pose using a magnetic tracking system and the tactile forces distributed on the inside of the hand are used for the object, a point cloud model is used, obtained from in-hand exploration, as shown in previous works [36], from RGB-D camera and also off-line from a laser scanner sensor. An online database, the Handle Project Data Collection Database [48], is publicly available with the datasets collected.

### *4.3. Learning object graspable regions: assigning weights to shape primitives*

In this section we address how to assign a weight (based on human statistics) to an object shape primitive for an initial grasping, ignoring for now the task context. The objective is to search for the shape primitive that has assigned more weight (between the three components of the object) to describe this region as suitable for grasping.

Through the human grasp demonstrations, we analyse the human choice to find the object graspable component given the three geometrical primitives that compose the object shape. We are biasing the geometrical primitives using the statistical data by quantifying the human grasp demonstration with a probability distribution based on histogram techniques. This way, given an unknown object and its three components, we will have weights distributed for each primitive to know which one is the best part as a candidate region for grasping. Afterwards, we have built a learned table with the information of primitives preferences based on their weights. Adopting a verification of dual-combination of shapes, we verify the object components weights to know which component

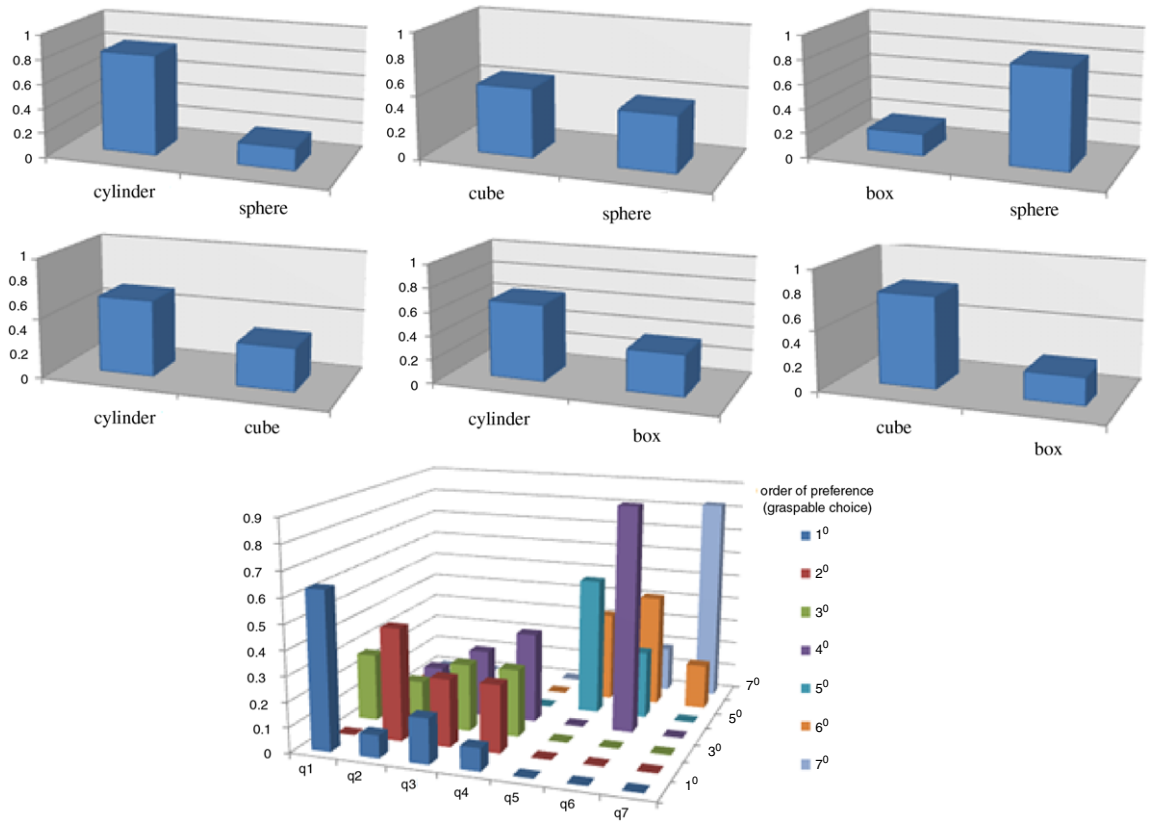



**Fig. 12.** Examples of the statistical data acquired during the human grasp demonstrations. The statistical data assists to weight the geometrical primitives as preference for grasping when dealing with a specific pair of quadrics.

is the graspable part, by comparing the weight of the first object component with the second, later the component with the bigger weight from the previous verification is compared with the third component of the object. The learned table from the observations is a probability table of a dual-combination of geometrical primitives, so that later an estimate can be made to select which part of the object is the best one for grasping when the grasp synthesis system faces different geometrical primitives on the object.

A heuristic rule is also used for biasing the geometrical primitive representing the bottom part of the object if its pose is in a vertical position, which decreases its probability of the most suitable region for grasping. If the object pose is in a horizontal position, then this rule does not apply to the bottom part, because it can be a candidate for grasping in the same way as the other object regions.

During this learning process, we analyse the human preference for grasping given a set of primitives that can compose an object. A questionnaire was made to find out the human's choice given two shapes primitives. A set of geometrical shapes was shown to the subjects, and they were instructed to point (grasp) the shape that is the graspable choice (easier to grasp or could be grasped in more different ways than the other one) in the subject's point of view. The primitives used to observe the human's choice are the set of the defined superquadrics for this work as described in Section 3.2.

The set of shapes were shown to each subject (demonstrated two shapes per time), and to facilitate the human choice, the subject could grasp and interact with both shape primitives, and later the subject had to decide which one is preferable for grasping, if those two shapes were part of a single object. A system was developed (questionnaire) showing the two shapes that were demonstrated, and the subject should register in the system his choice (primitive 1 or primitive 2). An incremental function was computed for each primitive to build a histogram distribution. Each subject registered the choice for all possible combinations of the set of defined primitives.

The learning is achieved given the shape primitives (also referred to as quadrics $q_i$) that compose an object, so that they are labelled when a graspable choice is made, for later computing the distribution of each labelled primitive. The histogram is computed as follows:

$$n = \sum_{i}^{k} \mathscr{H}_i, \tag{21}$$

where $\mathscr{H}_i$ is a function that counts the number of observations that fall into each of the disjoint categories $c$ representing dual-combination of the possible $q_i$ (e.g., given an object, its components in dual representation is $q_1$, $q_2$ and $q_i$, $q_3$, where $q_i$ can assume the form of $q_1$ or $q_2$); $k$ is the total number of categories and $n$ is the total number of observations. Then the normalization to compute the probability distribution is achieved by:

$$P(c_k) = \frac{n_k}{N}, \tag{22}$$

where $0 \leq c_k \leq 1$ (normalization of each category $c_k$, i.e., combination of quadrics); $0 \leq k \leq K-1$, where $K$ is the total number categories; $N$ is the number of observations; $n_k$ is the number of observations for each category; $P(c_k)$ is the probability of the $k_{th}$ category.

Fig. 12 shows some examples of the statistical data represented in histograms. In this figure we can see some histograms of pairs of quadrics and the distribution of preference between both. We can also verify an example of the choice (preference) for the graspable quadrics when given a set of quadrics.

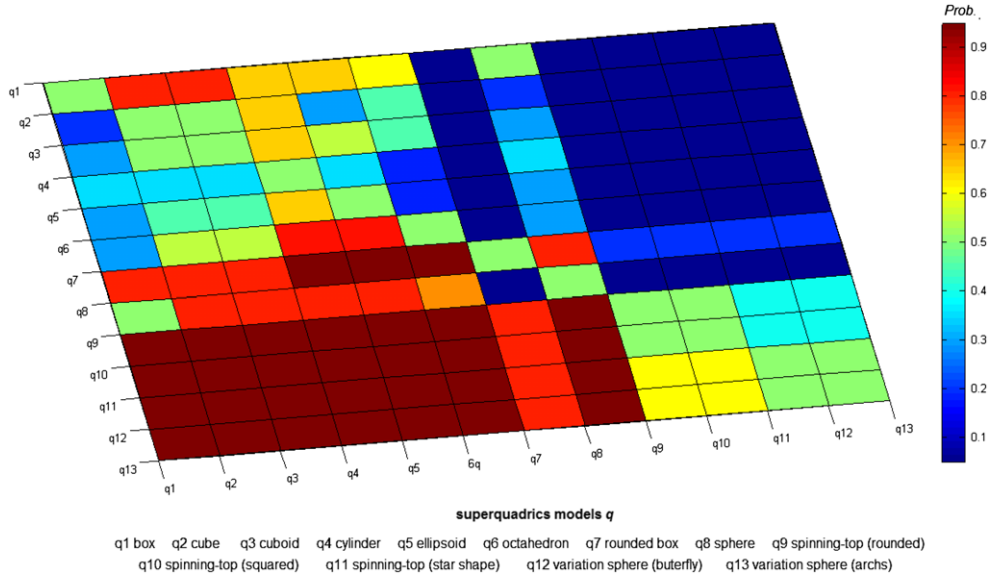



**Fig. 13.** Probability Distribution in the Learned Table: Pointing the preference given a pair of quadrics $q_i$–$q_i$, $i = \{1, \ldots, n\}$. The axes $\{x, y\}$ represent all possible pairs of quadrics $q_i$ that can compose an object. The probability assigned for the pair of quadrics is shown through the colour-map.

From these observations and statistics we could build a probability table (histogram), representing a learned table, as presented in Fig. 13. To read this learned table, the axes $\{x, y\}$ represent the possible pairs of the geometrical shapes (quadrics) that can compose a given object. The probability is assigned to the quadric $q_i$ in $x$ axis when it makes a pair with $q_i$ in $y$ axis. The colour-map varies from 0 to 1 representing the probability (weight) of each quadric $q_i$.

Later an inference for an object graspable region can be made, taking into account the quadrics that form the object, as demonstrated next.

#### *4.3.1. Inference for object graspable region given the object shape primitives*

We now address how to perform an inference to find the most probable graspable region on the object given a pair of primitives. The learned table presented in Section 4.3 (Fig. 13) is used as likelihood in a Bayesian inference to update the probability of a graspable region of an object given the combination of quadrics (e.g. given the sequence of the components/quadrics $q_i$ that compose an object: $q_1$ and $q_2$, $q_1$ and $q_3$, $q_2$ and $q_3$), we can identify the graspable region for an initial grasp type (not taking into consideration the task context).

The model for inference is given by:

$$P(Q = q_i|c_k) = \frac{P(c_k|Q = q_i)P(Q = q_i)}{\sum_j P(c_k|Q = q_i)P(Q = q_i)}, \tag{23}$$

where $P(Q = q_i|c_k)$ is the probability of the object graspable region $q_i$ given the combination (pair) of quadrics $c_k$. Then the classification is achieved according to the maximum a posteriori (MAP) estimate.

### *4.4. Learning suitable objects graspable regions in task-oriented grasps*

This learning process is based on our previous work [39] to identify graspable regions on objects that are suitable for grasping given a task context. We are also using this learning process for a more consistent estimate of graspable regions to assist the grasp synthesis when the system is under a task context. The idea is given an object model and task context, using the knowledge learned from human grasp demonstrations, the artificial system can estimate the best region (primitives) to perform a proper grasp for that situation. In this specific learning process, we have adapted our previous work with improvements for the grasp detection given the contact points on the object surface to quantify the human graspable choice.

In order to estimate the object region as graspable given a specific context (task-oriented), the combination of human demonstrations of stable grasps and object intrinsic information play an important role in the decision. In the learning process, we have the 3D object model of the object in a volumetric map, so that we can overlay the contact points of stable grasps on the object surface, represented in the grid cells of the object map. It also allows the identification of the grasp type by analyzing the contact points locations forming the hand configuration. The probabilistic representation of the object shape using a 3D map was proposed in a previous work and details can be found in [35,36].

For the human grasp demonstrations we are using from our experimental setup (Fig. 11) the finger 6D poses using a magnetic tracking system, and the tactile forces distributed on the inside of the hand. Thus, with the volumetric information of the object we can overlay the contact points given by human demonstrations on the object surface. The contact points locations are given as 3D positions of the fingers (acquired by the magnetic tracker sensors) when a subject touches the object (i.e. the tactile sensors are active). The contact points locations are easily overlaid on the object surface (cells in the object map), since we are working in the same frame of reference of the magnetic tracker allowing to map the contact points on the object surface. We consider that the system has previously acquired a 3D model of the object by in-hand exploration or other modality (e.g. vision) in order to have the volumetric model representation. Fig. 14 show examples of contact points overlaid on the objects surface. The figure presents an object grabbed by a human subject with a successful stable grasp during a manipulation task.

In this work, we can manually label the grasp type by observing the hand configurations during the grasp execution, or even in an automatic way, by computing the hand configurations using the contact points on the object surface. To automatically identify a

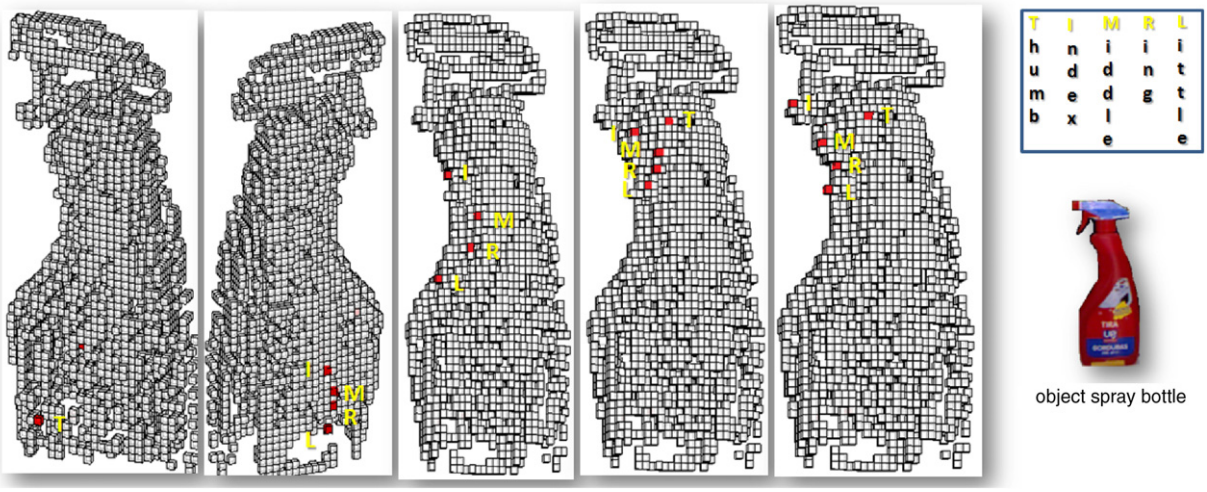



**Fig. 14.** Examples of contact points of stable grasps from human demonstration on the object (spray bottle) surface using the object occupancy grid (probabilistic map).

grasp type, we rely on the fingertip 6D pose relative to the wrist. For this, an additional tracker sensor is placed on the wrist. This allows us to compute the hand configurations defined in the grasp list [1]. To have the fingertip 6D data in the wrist frame of reference, we compute for each fingertip $f = \{x, y, z, roll, pitch, yaw\}$ (contact points) the following step:

$$f_{new} = \{(f_x - w_x), (f_y - w_y), (f_z - w_z), \alpha_w, \beta_w, \gamma_w\}, \tag{24}$$

where $f$ represents each fingertip in $\{x, y, z\}$ (in the frame of reference of the tracker sensor) and $w$ the wrist coordinates; $\alpha_w, \beta_w, \gamma_w$ are the angles roll, pitch and yaw of the wrist.

Later, the grasp type detection by using contact points on the object surface is achieved based on the transformed data representing a hand configuration. A grasp type is identified as a squared mean distance value between the fingers as shown in (26). As mentioned before, the grasp type used in this work are the ones defined in the grasp list adopted from [1], so that each discrete grasp type has an identity by using the hand configuration as explained next. First the Euclidean distances between thumb and index finger are computed, followed of the distances of the thumb and middle, thumb and ring, thumb and little, index and middle, middle and ring, and finally ring and little.

$$D_v = \frac{1}{N}\sum_{k=1}^{N}(d_{pq})_k^2, \tag{25}$$

where $d_{pq} = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$, and $p, q$ represent two fingers (thumb and index, and so on). Given a new observation of stable grasp, after computed the contact points distances, we can associate it to a pre-defined grasp by a similarity measure to search for the closest grasp:

$$\hat{g_s} = \min_{i\in\{1,\ldots,N\}} f(\hat{g_s}) = |D_v - \psi_i|, \tag{26}$$

where $D_v$ is the mean distance computed given the contact points; $\{\psi_1, \ldots \psi_n\}$ are the grasping thresholds, i.e., each grasp is represented by a $\psi$ value. Many observations (different subjects) of the same grasp value were computed (26) and an average for $\psi$ was achieved to represent a learned grasp. The result $\hat{g_s}$ is the grasp that will represent a new observation of contact points.

The grasp detection steps are presented in Algorithm 3. Fig. 15 shows some grasps types identified for an object used in the demonstrations.

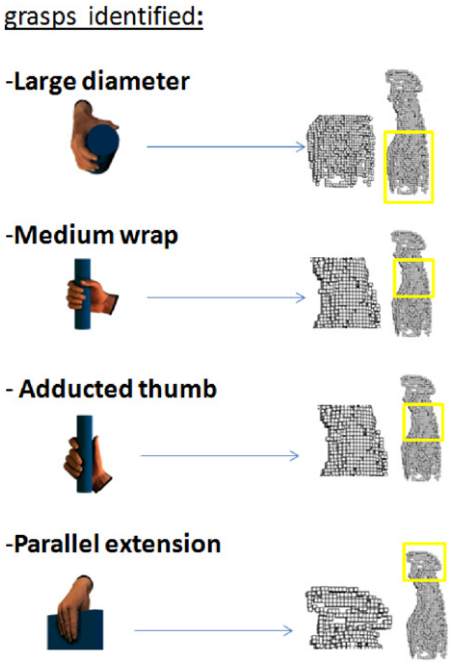



**Fig. 15.** Examples of grasps that were identified for the spray-bottle given the contact points (from human demonstrations) on the object surface.

After some trials of human demonstration on how to grasp an object given the objects models and the context, we could build a probability table to distinguish what kind of grasping is more probable to happen in each specific situation and also the object region that was chosen for the grasping.

Given a set of observations to represent a specific task $\mathcal{T}$, for instance, some simple tasks $\mathcal{T} \in$ {pick-up and place; pick-up and lift; pick-up and pour/tilt}, we have the probability of each grasp type in a specific context represented as $P(G|\mathcal{T})$. The probability of each grasp type $g_k \in G, k = \{1, \ldots, M\}$ in a specific context is given by the frequency of observations as expressed below:

$$P(G = g_k) = \frac{o}{M}, \tag{27}$$

where $o$ is the number of occurrences for the specific grasp type $g_k$ and $M$ is the total number of possible grasps $G$.

To identify the object graspable region, we verify the locations of the contact points on the object surface, and that region where

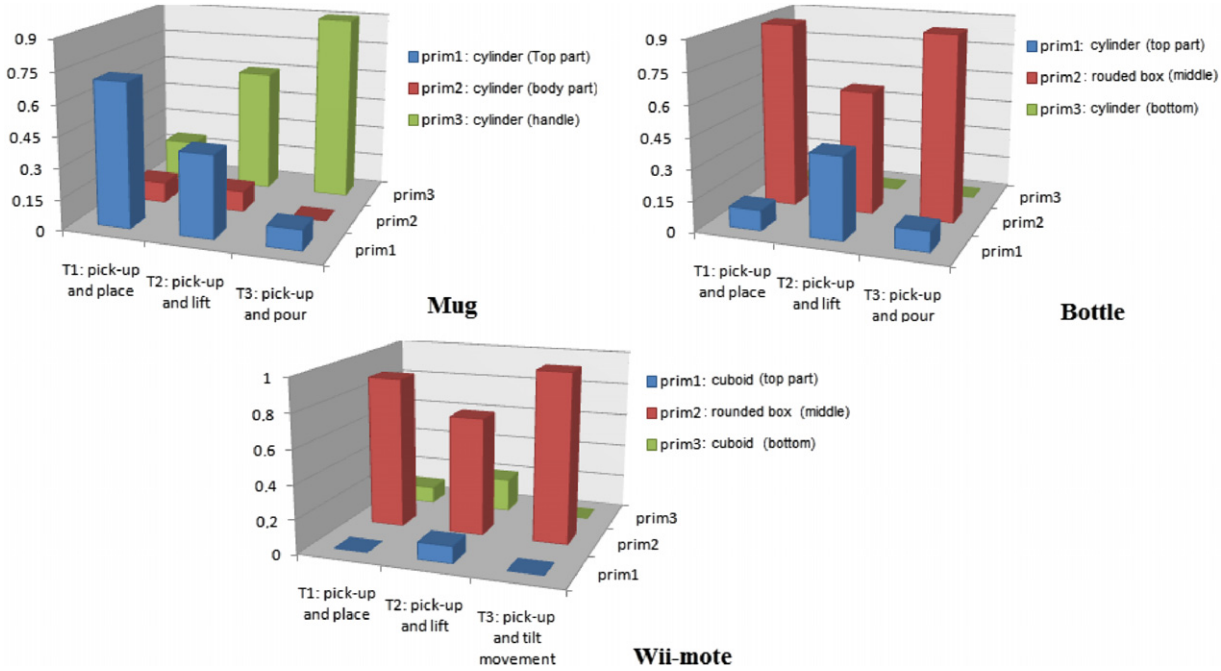



**Fig. 16.** Statistics computed from the observations. Three different tasks performed many times by 5 different individuals. By analyzing the probability distribution of the chosen primitives to perform the grasp, the object graspable part (given the task context) can be estimated.

**Algorithm 3:** Grasp type detection

1 Inputs: 6D fingertips contact points (acquired when the tactile sensors are active, i.e. touching the object)

2 For each fingertip: Compute a transformation of the fingertip 6D data $f = \{x, y, z, roll, pitch, yaw\}$ into the wrist coordinate system as demonstrated in Eq. (24);

3 Compute the Euclidean Distances $d$ between the thumb fingertip and the other 4 fingertips: $d(t,i), d(t,m), d(t,r), d(t,l)$, $t =$ thumb, $i =$ index, $m =$ middle, $r =$ ring, $l =$ little;

4 Compute the Euclidean distance between the other fingertips $d(i,m), d(m,r), d(r,l)$;

5 Compute the grasp value by an averaged sum of the squared Euclidean distances between the fingertips $D_v = \frac{1}{n}\sum_{k=1}^{n}(d_{i,j})_k^2$;

6 Search for the minimum distance $D_v$ and the learned grasping thresholds $\{\psi_1, \ldots \psi_n\}$, by computing a similarity function $\hat{g_s}$: $\min_{i\in\{1,\ldots,N\}} f(\hat{g_s}) = |D_v - \psi_i|$

7 Output: Grasp type

the points are located represents a quadric model $q_i$ (component of an object). Given a set of observations to represent a task $\mathscr{T}$, we have the probability of each object component being the object graspable region $P(q_i|\mathscr{T})$. It is computed in a similar way as shown in (27) where each component of the object has a probability associated with the graspable region given the context by computing the occurrences based on humans' choices for the object region defined as graspable.

Fig. 16 shows some statistics computed after the human demonstrations for the chosen object graspable component for a few everyday objects (mug, bottle and wii-mote).

*4.4.1. Inference for object graspable region in task-oriented grasps*

The object graspable region can be identified applying the Bayes' theorem. Given a task context $\mathscr{T}$, to identify the object graspable region between the primitives that compose the object $\{q_1, q_2, q_3\}$ as explained in Section 3, first it is necessary to detect the object components represented by quadrics models $q_i$. The probability distributions are obtained from the occurrence statistics acquired during the learning process used to build the likelihood.

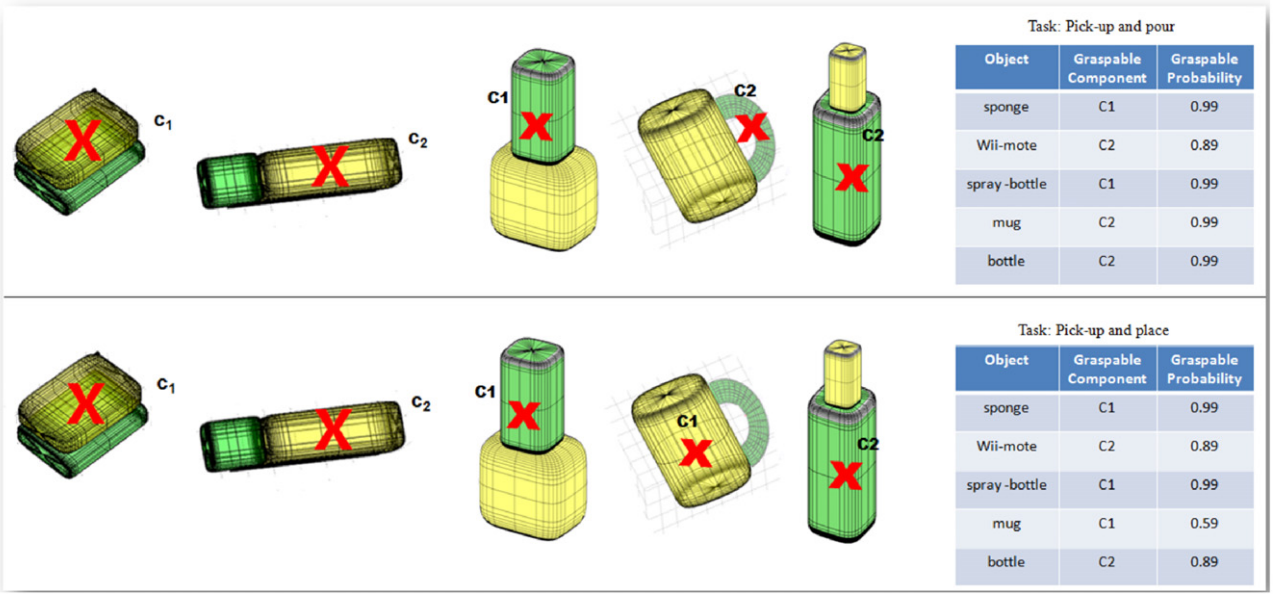
Given a context $\mathscr{T}$, we can estimate the object graspable part $q_i$ as follows:

$$P(Q = q_i|\mathscr{T}) = \frac{P(\mathscr{T}|Q = q_i)P(Q = q_i)}{\sum_j P(\mathscr{T}|Q = q_j)P(Q = q_j)}, \tag{28}$$

where the posterior information $P(Q = q_i \mid \mathscr{T})$ is computed for each primitive $q_i$ of the object in a specific task $\mathscr{T}$; the likelihood $P(\mathscr{T} \mid Q = q_i)$ is the learned probability for each primitive of the object given a task context as previously explained. The normalization factor is the sum of the probability of each object primitive being the graspable region.

A basic example of this application is given during the grasp planning, when a robot needs to execute a task. After detecting the object and its geometrical primitives, the robot can identify the object graspable region for possible suitable grasps, using the learned information from human demonstrations.

After learning a set of objects and task context, when the object is observed again in the same context, the system is able to detect the graspable part as shown in Fig. 17. The graspable component is chosen according to the maximum a posteriori (MAP) estimate. In these specific examples, we have used just two components due to two reasons: (i) the third component for these specific objects had zero probability or a very low probability assigned to the third component inside the specific context; (ii) these specific objects still have a good representation even with two components. Indeed, just for a better visualization of the results for these specific objects inside these specific contexts, we have adapted the results showing only the two more expressive components of the object. In a real situation, we keep the three components of the object, even if one of the components has a zero probability assigned to it.




Task: Pick-up and pour

| Object | Graspable Component | Graspable Probability |
|---|---|---|
| sponge | C1 | 0.99 |
| Wii-mote | C2 | 0.89 |
| spray -bottle | C1 | 0.99 |
| mug | C2 | 0.99 |
| bottle | C2 | 0.99 |

Task: Pick-up and place

| Object | Graspable Component | Graspable Probability |
|---|---|---|
| sponge | C1 | 0.99 |
| Wii-mote | C2 | 0.89 |
| spray -bottle | C1 | 0.99 |
| mug | C1 | 0.59 |
| bottle | C2 | 0.89 |

**Fig. 17.** Identification of the object graspable component for the sponge, *wii-mote*, spray-bottle, mug and bottle. For these trials we have used only two components for each object. Each component has a probability of being graspable, the maximum a posteriori estimate indicates the graspable component in each context.

In case of unknown objects, we have adopted a generalization process, reusing the prior knowledge for other contexts, for instance, if a unknown object has one primitive in common with a known object, a similar grasp can be attempted. The unknown object is in part matched a familiar object, i.e. after the object segmentation process, this object will have known geometrical primitives. Given a task, a Bayesian classification as shown in (28) is computed for each object primitive to infer the most probable object primitive for that task.

The feasibility and the quality of the work is somehow dependent of how a given object is represented after the segmentation and how its components are matched to a specific model. This way, the system can generate the hypotheses of regions on objects being graspable, and for each primitive a set of grasp types is associated.

### 4.5. Learning grasping choice from human observations

From human grasp demonstrations we can also observe the grasps types that are assigned to the object regions. This way, we can build a set of possible candidate grasp types to a specific object or for specific geometrical primitives that compose this object.

The learning process is achieved given a dataset with labelled examples of grasps types associated with an object component. Since we have the grasp detection and object components detection, we can learn and associate a set of grasps with each geometrical primitive that can represent an object component.

Through histogram based learning, we quantify the probability table containing a set of grasps for each quadric. We have observed some grasps by human demonstrations for the defined geometrical primitives. It means that $n$ grasps can be mapped to a specific shape, i.e., $G = \{g_k, \ldots, g_m\} \mapsto q_i$. Afterwards, a selection of the more probable candidates grasps for each geometrical primitive based on the probability distributions can be made.

The probability table that was built in this learning process can be seen as a 3D histogram, where each quadric $q_i$ (in axis $z$) has $n$ grasps $G$ (in axis $x$) with probabilities associated with each grasp (axis $y$) to be the most probable grasp for each quadric $q_i$. In this learning process, for each random variable, the distribution was normalized given the respective occurrences computed in a similar way as previously shown in (27).

In our framework we use taxonomy of grasping with a set of 33 grasp configurations. Since for many everyday activities the number of distinct objects is limited, we can have some non-observed grasp types. This happens because some of them were designed for more complex tasks of in-hand manipulation or fine movements, and not only for the initial grasp that is our focus here. In hand grasp transitions would require many more observations to achieve a representative distribution. In our learning approach, using a histogram-based technique, some features might have zero probability, because they have never been observed, i.e., a few grasps were never applied to some specific shape. These cases can be dealt with using the prior state of knowledge described by the rule of succession (Laplace's law of successions). The knowledge is given as an enumeration of the possibilities, with the additional information that it is possible to observe each category. The idea behind it is to assign a minor probability to the non-observed grasps avoiding a zero probability. Whenever these features with zero probability occur in the classification step, the correspondent hypothesis will receive also a zero probability. Since for the inference, the classifier is continuous, based on multiplicative update of beliefs, when the features from the likelihood with zero probability are used, these zeros might lead to a definite out-rule of hypothesis. To avoid this problem, a minimum probability for non-observed evidences is produced using the rule of succession as indicated:

$$\forall n_i = 0, \quad P_{\min}([n_i = 0]) = \frac{n_i + 1}{N + \chi} = \frac{1}{N + \chi}, \tag{29}$$

where $P_{\min}([n_i = 0])$ is the resulting minimum probability that will be assigned to the non-observed grasping ($n_i = 0$); $\chi$ represents the total number of features (i.e., all possible grasps types, $G = 33$); $n_i$ is a specific feature (in this case $n_i = 0$, the non-observed grasps); $N$ represents the total of occurrences (sum of all features occurrences).

The probability table gives us the likelihood useful for estimating both $P(G|Q)$, probability of a grasping occurring given a geometrical shape, and $P(Q|G)$, probability of a geometrical shape given the set of grasp types. However, in our specific case, we rely on the statistical data to associate the most probable grasps with each quadric, using their probabilities as weights to set the preference of the candidate grasps.

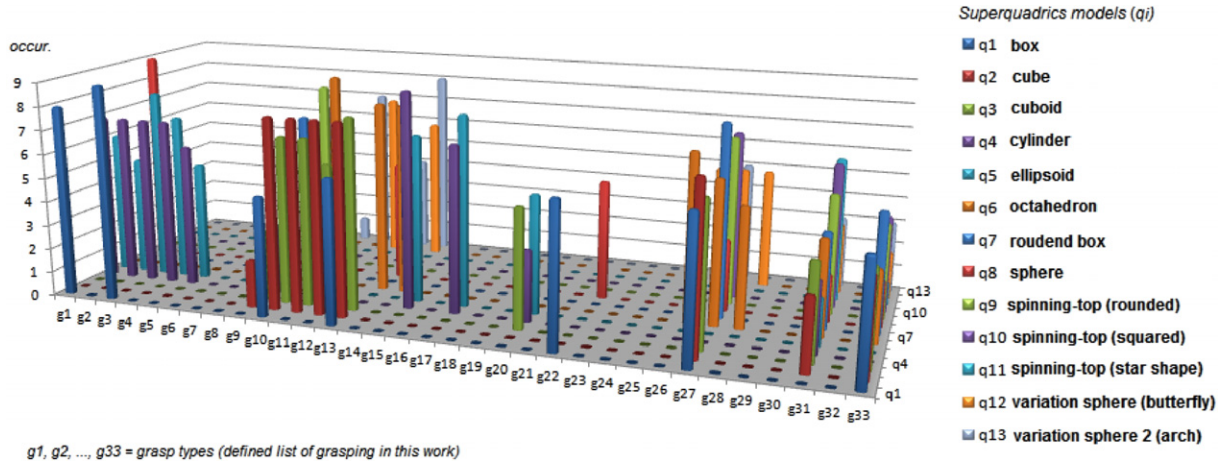



**Fig. 18.** Grasp choice given the object quadrics: statistical data acquired by human demonstrations. The demonstrations were chosen from a grasp list [1] with 33 grasp types for 13 possible quadrics models Q = {*box, cube, cuboid, cylinder, ellipsoid, sphere, octahedron, rounded box, rounded spinning-top, squared spinning-top, star spinning-top, variation 1-sphere(spherical arch), variation 2-sphere (butterfly shape)*}.

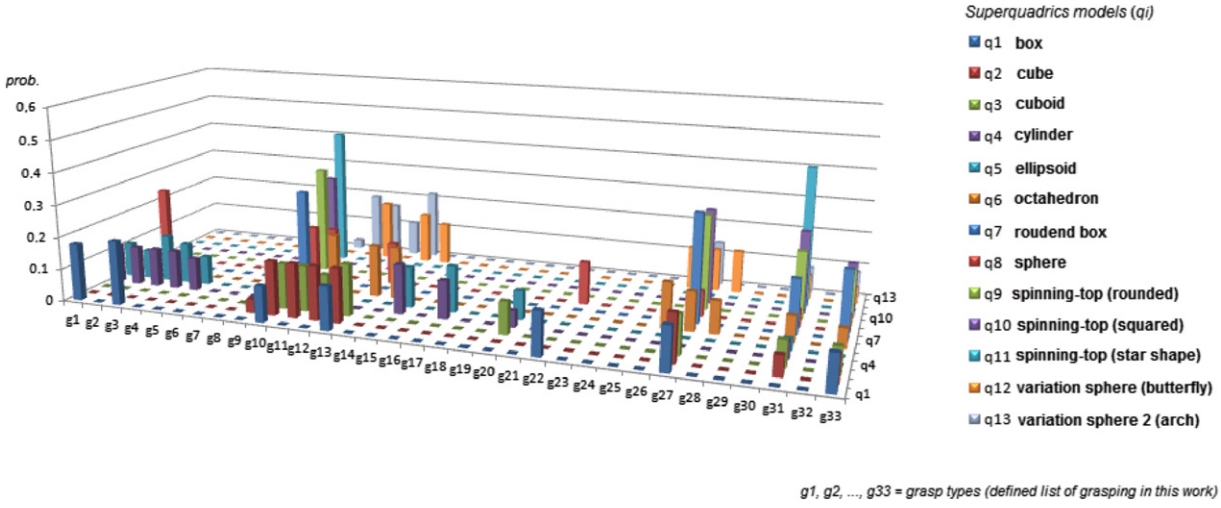



**Fig. 19.** Probability distribution: learned table from the statistics presented in Fig. 18. Each grasp type has an occurrence probability given a quadric model.

Fig. 18 shows the raw data representing the statistics from human demonstrations where 10 subjects, 9 man and 1 woman, all righted-hand, aged between 22–33 years old, demonstrated for each quadric the most probable grasps (from the grasp list [1]) with a minimum of 1 up to 10 grasps. The 10 subjects have demonstrated for the defined superquadrics models of this work, a total of 510 demonstrations (possible grasps). From this data we could verify different statistical information, such as the preferences and the mode of the samples. For instance, the superquadrics models that had more associated grasps were: cylinder, box, cube, cuboid, sphere. The grasps with more frequency during the demonstrations: g27-quadpod, g13-precision sphere, g1-large diameter, g3-medium wrap, g31-ring.

Fig. 19 shows the learned table with the probability distribution after a normalization of the statistical data, which is useful for inference. The normalization for the likelihood is achieved by $P(Q|G = g_k) = \frac{g_k^o}{M}$ where $g_k^o$ is the occurrence of a specific grasp $g_k$ during the human grasp demonstration and $M$ is the total of demonstrations (all possible grasps) for a specific quadric $q_i$.

#### 4.5.1. Inference for grasping choice

The inference over the learned table presented in Section 4.5 (Fig. 19) is represented by two possible questions: first $P(G|Q)$ meaning the probable grasp given one or more quadric model representing an object, and second $P(Q|G)$ meaning the opposite, the most probable quadric model given a set of grasps. The first and second inference are computed adopting Bayes rule since we know the likelihoods and priors. The Bayesian inferences are done as follows:

$$P(G = g_k|Q, s) = \frac{P(Q, s|G = g_k)P(G = g_k)}{\sum_j P(Q, s|G = g_k)P(G = g_k)}, \tag{30}$$

$$P(Q = q_i|G, s) = \frac{P(G, s|Q = q_i)P(Q = q_i)}{\sum_j P(G, s|Q = q_i)P(Q = q_i)}, \tag{31}$$

where $s$ represents a set of information (temporal), for instance: for (30) a set of quadrics to update the probability of the candidate grasps for an object composed of $n$ quadrics; for (31) a set of grasps to update the probability of the possible quadrics.

The prior $P(G)$ in (30) is obtained from another distribution function (probability table) as presented in Section 4.3. The prior $P(Q)$ in (31) is a uniform distribution.

### 4.6. Learning from object observations

More information is extracted when dealing with the object model. Through observations of object components, after the

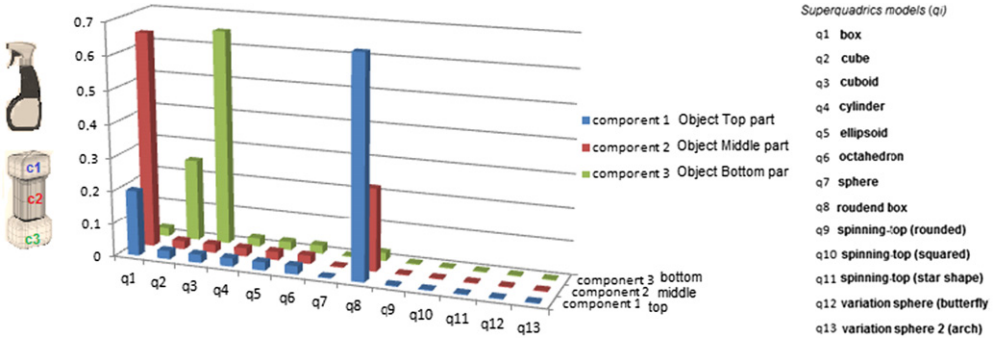



**Fig. 20.** Spray bottle — probability distribution of $q_i$ being considered as an object component $c_i$.

detection given a point cloud $\mathbb{P}$ from a specific sensor, we also learn and build a probability table by analyzing some statistics. Adopting the same strategy of histogram based learning as explained before, we have built for each everyday object, a probability distribution taking into consideration the components of the object. Given a set of quadrics $Q$ we have the probability distribution of each quadric $q_i$ being the component of the object (i.e., belonging to an object region, such as the top, middle or bottom).

Fig. 20 shows an example of a probability table of a specific object (spray bottle) demonstrating the probability of each quadric $q_i$ being an object component. The same was done for other objects.

Later an inference can be made to identify an object as demonstrated in the next subsection.

#### 4.6.1. Inference for object identification

The inference to identify an object given the sequence of quadrics $q_i$ following the order $\{q_1, q_2, q_3\}$ is computed for all variables (i.e., all possible objects identities) using (32), and each one has a likelihood (Fig. 20) representing the learned components of an everyday object. The identification is computed as follows:

$$P(O|Q, s) = \beta P(Q, s|O)P(O), \tag{32}$$

where $P(O|Q, s)$ is the probability of an object identity given the sequence of quadrics representing each object component; $\beta$ is the normalization factor, $\beta = \frac{1}{\sum P(Q,s|O)P(O)}$ meaning the sum of all likelihoods (learned table for all objects) for those 3 object components detected; $s$ represents the sequence $\{1, 2, 3\}$ of the detected quadrics.

With the learned information and using the inference for a limited set of objects, we have built a table containing the inference results, so that when the artificial system faces a novel object, it can detect a combination of 3 quadrics to identify the object or at least reasoning that it might be similar to one previously observed. Identifying an object is another alternative to find a graspable region, as well as grasp associated with this object or its components to find the candidate grasps.

### 4.7. Storing learned data

Since we have a limited number of objects, object components modelled with superquadrics models, task context and grasping types, we can restrict all possibilities of one or more random variables using the inference results and the learning data.

We have built tables storing the learned data, as well as tables with the inference results for the set of possibilities from the learned data. We can use this information to generalize, and apply in other contexts, or in case of grasping or objects, we can use similarities, i.e., find the most similar one to apply in a new context or to an unknown object.

This process was done in order to have sufficient data stored to facilitate and speed up the processes for a real time application during the execution, reducing then the processing time, since we want a system working in a feasible time for our application that can also be incorporated in other context (e.g., for in-hand manipulation tasks with grasp transitions).

## 5. Grasp synthesis

The main purpose of the grasp synthesis is to find feasible grasps, given a 3D object model, and the robot end-effector configuration to maintain a stable grasp during the execution. Therefore, to accomplish this task, we first need to use the object characteristics to find the proper region for grasping, as well as the pose and configuration of the hand relative to the object to approach the object and successfully grasp it. Thus, we have developed an artificial system based on the idea presented in Fig. 1. The experimental robotic system was a joint effort built within the HANDLE project consortium [49], and ROS (Robot Operating system) was used to combine contributions from all the project partners to have a working system. This work is one of the contributions integrated in the project. In the next subsections the modules that comprise the grasp synthesis system architecture are presented, as well as the details of the system implementation.

### 5.1. Using decomposition module in the grasp synthesis

The decomposition is used for the grasp synthesis objective. As explained in Section 3, given an object point cloud (unknown object), we first decompose the object into key components and later, inferences on the learned data to detect suitable regions for grasping and the proper configuration are made, as explained in Section 4.

In this work we have used as a pre-processing step in our system a ROS module named Extract Objects and Table developed by the HANDLE project consortium [49]. When the object point cloud is acquired by an RGB-D sensor, by using the ROS algorithms from the PCL (Point CLoud Library) [50], we can extract the segmented object point cloud, removing the table and background. The object for manipulation is placed on the table in the sensor range. Algorithms like RANSAC (Random Sample Consensus) [51] are used to remove the tabletop, removing the non-interesting regions of the point cloud resulting only the object point cloud or a cluster of objects in case of many objects.

Another important pre-processing addresses the problem of the partial object view provided by the RGB-D sensor. Within the ROS HANDLE modules, a PCA-based method is used to fill out the object partial view into a symmetric shape, passing to our system a point cloud of the complete object. PCA provides the main axis of the point cloud, upon which an ellipsoid fitting is made to enable filling out the object, providing a final object point cloud using an object centred frame of reference. The major axis, the one with

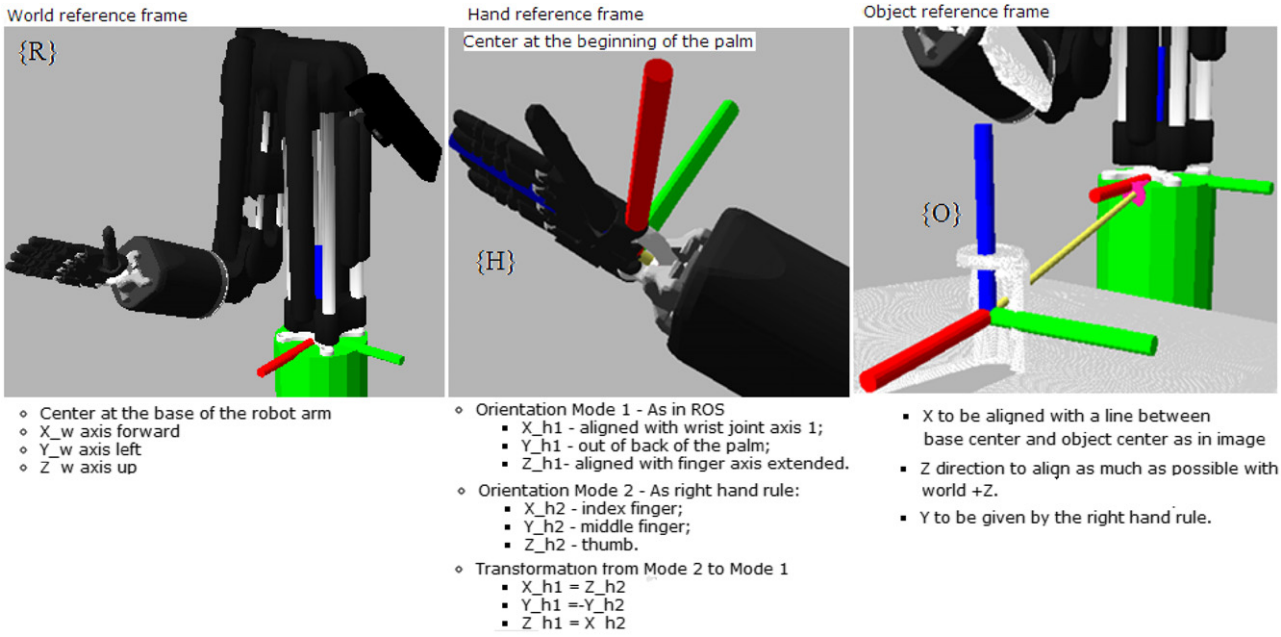



**Fig. 21.** Frame of references adopted to generate the grasps pose relative to the object pose.
*Source:* Figure adapted from HANDLE Project Wiki page for the definitions of the ROS modules for the final demonstration of the project [49].

larger magnitude, will be the axis along which we perform our segmentation.

Following the ROS structure, the nodes communicate between each other by publishing messages to topics. A message is a simple data structure including types or arrays similar to the structures defined in C/C++ programming. The nodes were implemented in C++ (OOP — Object-oriented programming) adopting the ROS architecture. The object point cloud is then passed as a message to the Decomposition module. Then in the segmentation step, the object is converted into a new frame of reference (object-centred). The inputs for the second node of the Decomposition module (shape approximation) is a table of object segments (e.g., the segments $\mathbb{P}_{top}, \mathbb{P}_{mid}, \mathbb{P}_{bot}$) that was published as a message to a specific topic by the first node (segmentation). The output of this second module is a published message into a topic containing the 15 parameters of the superquadrics model, representing the scale in each axis $\{a_1, a_2, a_3\}$, two parameters representing the superquadric shape $\{\in_1, \in_2\}$, three parameters representing the translation $\{p_x, p_y, p_z\}$ and three angles representing the rotation $\{\phi, \theta, \psi\}$ in each axis, as well as the centroid coordinates $\{c_x, c_y, c_z\}$ and the volume of the quadric $v_q$.

In the second node, since we obtained the object pose and size by the computation of the superquadrics models, we can use the extracted information to generate the candidate grasps for each quadric $q_i$ of the object based on the learned data.

### 5.2. Grasp synthesis module

This module is in charge of searching for the proper candidate grasp, returning a list of grasp hypothesis given a 3D object model, as well as the correct hand pose to approach the object for grasping. All learned data that was stored are used in this module to assist the grasp generator to make inference over the data, as mentioned in Section 4. In our approach we decided to store the learned data and some inference results over some pre-defined situations such as grasp types associated with some geometrical models to gain time, reducing in this way the processing time. Later with the grasp type for a given object, we have to compute the hand pose relative to the object for the grasping execution.

The developed artificial system will always face an inference given the object information, these possible inferences were previously described in Section 4, which demands the use of the learned data from human grasp demonstrations for the estimate. The next subsection will present more details on the grasp list generation.

#### 5.2.1. Grasps list generation

When the artificial system receives the inputs coming from the decomposition module, the objective is to then have the candidate grasps for each part of the object. The superquadrics parameters, the object centroid, object pose and scale (in the metrical superquadric coordinate system) are computed as presented in [46]. This way, we know the object orientation and the limits of the object (width, height and depth), which allows the system to generate the possible candidate pre-grasp and the grasps near to the object boundary.

A discrete space-state is used as defined in the HANDLE project [49] for the frames of references (world for robot platform base, robotic hand and object) as presented in Fig. 21. The hand approaches the object based on the object boundary, computed using the object centroid and the superquadrics parameters (size). The hand pre-grasp is denoted as the initial robotic position, usually behind the object (robotic platform point of view). The distance from the pre-grasp is computed from the origin of the hand (centre of the palm) to the object boundary. The natural hand configuration (open hand) is the pre-grasp (one state before the selected grasp configuration).

Having computed the object pose during the decomposition module, the system then searches for the possible candidate grasps for the detected superquadrics. Afterwards, for each grasp, the hand pose relative to the object is computed at pre-grasp (neutral state before grasping) position and also for the selected grasp after inference. We set the pre-grasp position of the hand away from the object (behind it) with neutral state (open hand) as shown in Fig. 21.

The hand pose for each grasp type is computed for the top and side-grasp orientation relative to the object pose. The approach direction is dependent on the graspable region that was chosen and

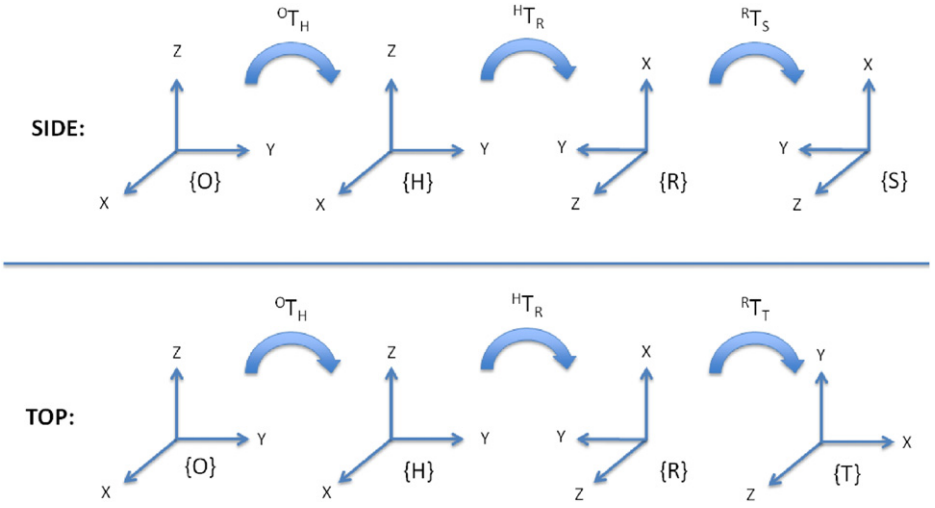



**Fig. 22.** Transformation diagrams for getting the grasp pose in relation to the object frame of reference.

taking also into consideration the object size and if it is symmetric in its shape. Then the approaching direction on the object will be on its top or side direction around the centroid of the chosen component for grasping. The hand kinematic is also important, e.g., when the graspable choice is on side-grasp orientation for a specific grasp type, if the object is symmetric the approach can be performed on the right side of the object or in its frontal side (robotic platform perspective). The approaching direction is also dependent of a task context; usually the top-grasp pose approaching direction is the chosen one for simple tasks like pick-up and place. Some grasps are limited to the side position, such as adducted thumb or medium wrap, used for example, to grasp a pipette by its side when it is placed on a vertical stand.

To generate the hand pose in top and side positions, the object pose and size, the frames of reference of the object and the hand are taken into consideration. The robotic platform consists in several joints and links as seen in Fig. 21 (Shadow dexterous Hand [2]). A proper frame of reference structure was defined and thus the relations between the Shadow dexterous hand and object need to be calculated. In Fig. 22, the relevant frames of reference is specified and the transformation details are presented. The final goal for getting the transformation matrices was to be able to set the correct grasp pose of the hand relative to the object. The following expressions define the side and top grasp poses transformations respectively:

$$^{O}\mathbf{T}_{S} = {}^{O}\mathbf{T}_{H}\,{}^{H}\mathbf{T}_{R}\,{}^{R}\mathbf{T}_{S}, \tag{33}$$

$$^{O}\mathbf{T}_{T} = {}^{O}\mathbf{T}_{H}\,{}^{H}\mathbf{T}_{R}\,{}^{R}\mathbf{T}_{T}, \tag{34}$$

where $^{O}\mathbf{T}_{H}$ defines the transformation between the frames of reference of the Object $\{O\}$ and a hand wrist $\{H\}$; $^{H}\mathbf{T}_{R}$ defines the transformation between the frames of reference of the hand $\{H\}$ and the actual frame of reference used in the robotic platform $\{R\}$ for the hand, as illustrated in Fig. 21; $^{R}\mathbf{T}_{S}$ defines the transformation between the frames of reference of the robotic platform $\{R\}$ and the side grasp position $\{S\}$; $^{R}\mathbf{T}_{T}$ defines the transformation between the frames of reference of the robotic platform $\{R\}$ and the safe top grasp position $\{T\}$.

Below are the matrices that relate to these frames of reference:

$$^{O}\mathbf{T}_{H} = \begin{bmatrix} & Rsq & & 0 \\ & & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{H}\mathbf{T}_{R} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$^{R}\mathbf{T}_{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a_2 \\ 0 & 0 & 1 & -\Delta W_H \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{35}$$

$$^{R}\mathbf{T}_{T} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & a_3 \\ 0 & 0 & 1 & -\Delta W_H \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $a_2$ and $a_3$ are the dimensions of the superquadrics that is necessary to set the grasp pose away from the object at a certain distance, and $\Delta W_H$ is the distance from the wrist to the centre of the hand. $R_{sq}$ represents a rotation matrix based on the $\{\phi, \theta, \psi\}$ (yaw–pitch–roll) angles extracted from the superquadrics components.

The grasp list is given by the grasps associated with each quadric $q_i$ that composes the object as previously detailed in Section 4. After generating the grasp list and their poses for top and side approach, the artificial system will choose using a rank pool of weighed grasps, based on the learned probabilities. A high weight is assigned to the grasp in case of a success in a specific context, if a failure happens, the grasp will have a lower weight assigned to it in that context. Simulations were done a priori to test specific grasps in different hand poses and different contexts to assist the grasp rank pool to update the weights of each grasp in specific situations and with specific objects. The decision and learning of the rank pool module is based on Gaussian Process Regression implemented and applied by a partner of the HANDLE project consortium as presented in [52].

The grasp execution by the robotic platform is performed after the acquisition of the object point cloud and subsequent processing for the grasp generation. The grasp generator module encloses other modules beyond the scope of this work. In fact, our module is one of three that compete to provide suitable grasps into a common rank pool from which a decision is made. The system also has a GUI (Graphical User Interface) to monitor and also interact when necessary to remove candidate grasps or take decisions.

For the robot execution (which it is not the main focus of this work), the mapping of the chosen grasp to the robotic platform takes into consideration the hand pose of the grasp, object pose and the robotic hand kinematics. The adopted approach was developed within the HANDLE project consortium to map the grasps to the robotic hand, and implemented an *Eigengrasp-planner* [53] using the GraspIt! simulator [37]. It explores the use of grasp-synergies and uses the *eigengrasps* for the Shadow robotic hand [2]. The basic

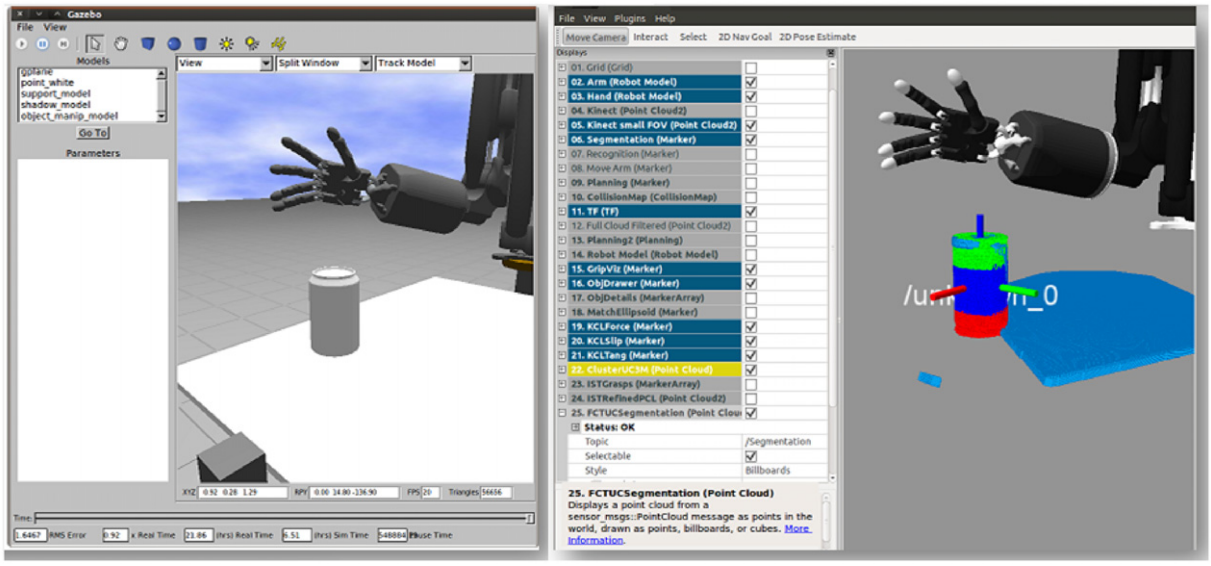



**Fig. 23.** Decompose view in a simulator. The images show all steps for the segmentation. Left image represents the raw object data from the Kinect sensor after removing the table-top. The right image shows the segmentation result of the unknown object achieved by our decomposition module.

**Algorithm 4:** Grasp synthesis steps before the robot execution

1 Input: Object Point Cloud $\mathbb{P}$

// Object Decomposition:

2 *pclouds*[$\mathbb{P}_{top}$, $\mathbb{P}_{mid}$, $\mathbb{P}_{bot}$] ← `segment` ($\mathbb{P}$);

3 *objParamList* [$a_1, a_2, a_3, e_1, e_2, x, y, z, \phi, \theta, \psi, c_x, c_y, c_z, v_q$] ← `getShapeSQ` (*pclouds*);

// Using the stored Learned data for inference given the object shape parameters:

4 *grasps* [] ← *getGraspList* (*objParamList*);

// Generating Hand Pose (grasp list) relative to the Object Pose

5 handPoses [] ← *genGraspPose*(*grasps*, *objParamList*);

6 Outputs: Grasp List (*grasps*) and their poses (*handPoses*) relative to the object ($\mathbb{P}$)

idea of grasping based on synergies is to combine a quick search of the reduced subspace spanned by the relevant *eigengrasps* with a later adjustment phase as a hierarchical approach, where the synergies pre-shape the hand with approximate finger positions around the object. Sampling a large set of suitable (e.g. human-like) hand poses and performing a principal component analysis, the resulting set of eigenvectors provide a new basis of the hand joint space, where the set of eigenvectors defines the synergies matrix. Further details are given in Deliverable 24 (D24) of the HANDLE project [54]. For each discrete grasp type used in this work, a mapping is made using this strategy, allowing later the correct grasp execution by the robotic hand.

Algorithm 4 presents the general idea of the grasp synthesis. The algorithm uses all functions demonstrated in Sections 3–5 following this order respectively: Decomposition (object segmentation and shape modelling) and Grasp List Generation (candidate grasps given the object parameters and their poses relative to the object). The algorithm does not enclose the execution part, where the *egeingrasps* are used to map the discrete grasps to their correct configuration for the robotic hand performing the object grasping, since our goal is the grasp generation.

## 6. Experimental results

As previously mentioned, the artificial system was implemented under the ROS platform using C++ language to perform the grasp synthesis. The sensor used to acquire the object point cloud during the robot execution is an RGB-D camera. The processing time of the artificial system to run all algorithms described in this work for grasp synthesis (Decomposition and Grasp Generator) takes on average up to 2 s before executing the grasp. From the all algorithms, the most time consuming is the shape modelling using superquadrics, since it depends on the size of the point cloud to compute the parameters. The learned data and the pre-processing inference enable us to have a fast decision over the object model. The next subsections show the results achieved using our proposed artificial system for grasp synthesis.

### 6.1. Simulated tests

The first stage of tests of our application were performed in an off-line mode. Basically, we have simulated an application that triggers all modules (Decomposition and Grasp Generator), passing as input a point cloud previously acquired from different sensors. This way we could verify the consistence and the outputs of the system for those objects.

Fig. 23 depicts the segmentation of an everyday object (unknown to the system). First image (top-left) shows the data (raw object point cloud) from the sensor (MS-Kinect) after removing the table-top. The left image shows the segmentation of the object during the decompose module.

Fig. 24 shows some everyday objects just to exemplify the outputs of each module presenting the candidate grasps and hand pose (in top and side grasp orientation for the chosen grasp type) of the object graspable parts. The generated grasp list follows an order of appearance indicating the grasp with highest weight (higher probability to be the selected grasp) down to the lowest one. The figure presents the result of our simulated tests for those objects.

Some assumptions need to be taken into account before sending the list of candidate grasps to the next module. We have to verify if the grasps are feasible for specific parts of the object. For instance, if the graspable part is the middle component of the object, then, top-grasp orientation is not allowed for that component if it is in a vertical position, and the same happens to the bottom part. Some

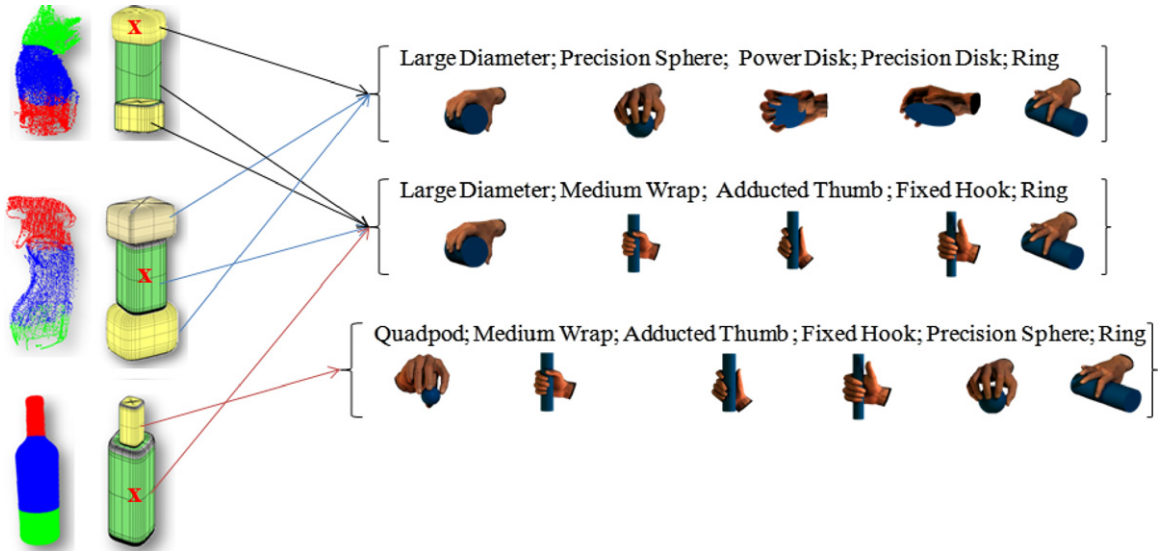



**Fig. 24.** Results using the object point clouds to test our modules. The application returned some grasp associated with the geometrical shapes (quadrics $q_i$) of the object. The marked quadrics in red are the object parts with higher probability to be the graspable part, and the grasps associated with this specific part have a higher weight. The order of appearance of the grasp types indicates the most probable grasp for that component.

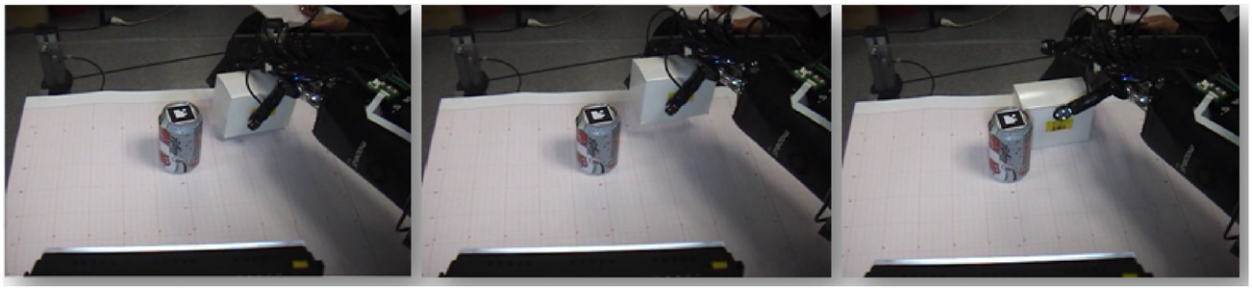


**Fig. 25.** Selected grasp (grasp 27 from [1]: Quadpod) for the object (box) executed in a robotic platform.

grasp configurations for the top and bottom parts of the object may be not proper, due to the size of the segmented part. So, after generating the grasp list, the system follows some heuristic rules to discard the non feasible grasps.

In general, the results achieved from our simulated tests are suitable for the objects presented to the system.

### *6.2. Tests in the robotic platform*

The modules explained in this work are used to search for feasible grasps given a 3D object, these are then mapped to the robotic platform using the correct kinematics for the execution. Here, we are not dealing with the planning of the trajectory to approach the object, only the grasp type (and its pose) for the robotic platform. The trajectory planning (reaching movements) are addressed by other modules of the integrated system inside the HANDLE consortium [49].

The criteria that the robotic platform uses to chose a specific grasp was defined inside the HANDLE project [49] and it is explained as follows:

- If one of the provided grasps is a good starting point to reach the final grasp after in-hand manipulation that is set by the task (grasp transition sequence is easy);
- If the quality of the grasp is higher than those ones provided by other means (other grasp generators);
- If the grasp candidate (hand pose) is suitable for the Inverse Kinematics limits.

In general our modules provided good candidates that assisted the robotic platform in performing successful grasps. However some problems were encountered, such as the robotic hand pose not being reachable, because it is so close to the table, and finding a correct offset of the grasp to avoid the hand colliding with the object before completing the grasp. More trials will enable a better tuning and also to re-weight the candidate grasps based on the success rate.

Fig. 25 shows an example of execution when one of the grasp of our modules was chosen as suitable for the object. The sequence shows when the robotic hand touches the object to perform the grasp, then the robotic hand lifts the object and finally releases it.

Fig. 26 presents the simulation (inside the integrated software developed under HANDLE consortium [49]) to chose a specific grasp before the robotic platform execution. In this specific case our grasp was chosen and tested to validate the grasp.

Fig. 27 shows a sequence of tests using the grasps from our system to indicate which grasps are valid and not valid.

The everyday objects used to test the modules are unknown to the system, so that we are applying the mentioned modules to approximate the object shape into familiar shapes to generate a set of candidate grasps for each known shape. In general, we can state that, after some improvements using offsets for hand pose to avoid some restrictions (table limit, hand kinematics), the grasp generator module is a solution to generate valid grasps for everyday objects.

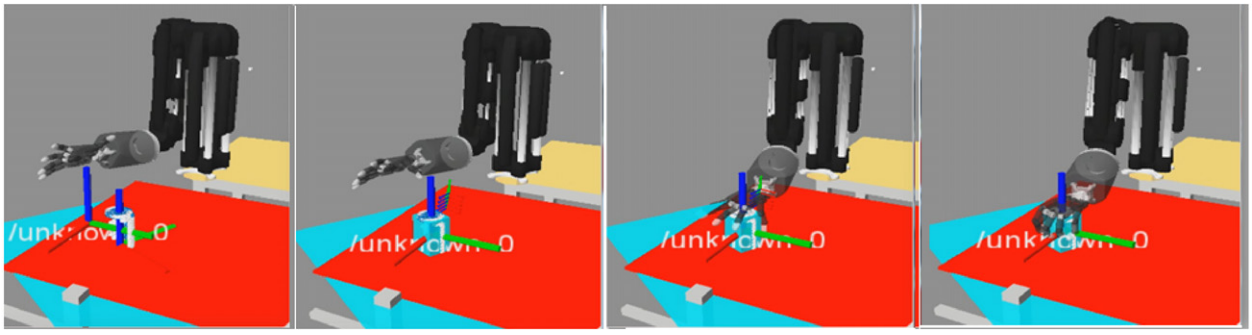


**Fig. 26.** Selected grasp executed in a simulator before the execution in the robotic platform.

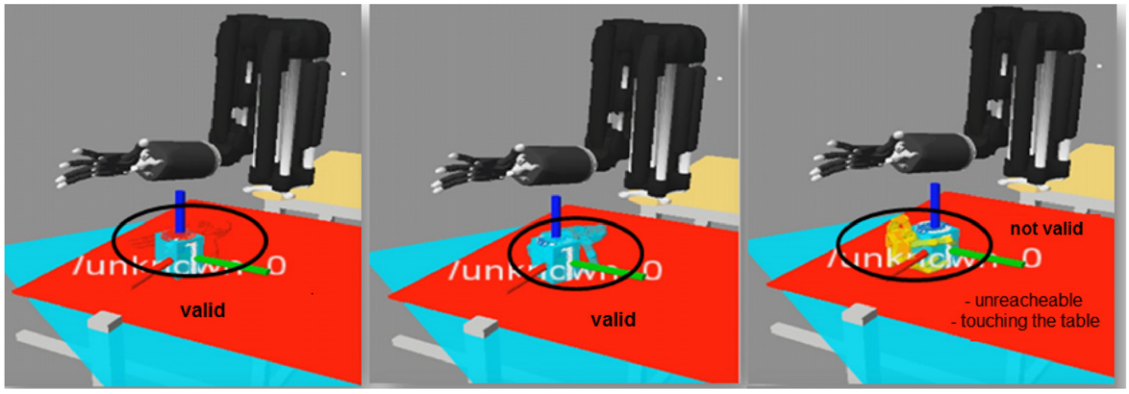



**Fig. 27.** Tests in a simulator to verify which grasps from the generated list (hypothesis) are valid.

## 7. Conclusion and future works

An artificial system for grasp synthesis given a novel object to be manipulated by a robotic dexterous hand was presented. The approach follows a novel combination of knowledge acquired from human demonstrations, with a simple breakdown of the object into parametrized quadrics that allow matching with the previous knowledge. The focus is on having a fast synthesis of ranked viable grasps, suitable for a real time robotic implementation, avoiding time consuming object shape analysis. Given an unknown object, the corresponding point cloud is segmented into components, and these are approximated by superquadric models. Previous human demonstrations with know objects, also broken down into components, enable learning the viable grasp probability distributions associated with the superquadric primitives, across which a matching is done to make an inference to obtain grasp candidates for the new object. The proposed approach limits the amount of grasps for each known object primitive based on the learned human choices. The implemented system generates a list of candidate grasps for the new object providing a ranked pool of possible grasps.

As detailed previously, other approaches in the literature use object decomposition into parts, and some also use superquadric models. However our approach focuses on having a fast ad-hod segmentation solution, that is sufficient since it is combined with knowledge from human demonstrations. Other approaches that also rely on human demonstrations manually label the object regions and grasps, our system relies on and automatic process. The use of the probabilistic approach, together with a target robotic dexterous hand that uses finger synergies to adapt the grasp completions, makes our approach effective, even if less precise modelling is done relative to other computationally heavier solutions.

Our approach was integrated into a full robotic grasping system, and simulated and real experimental results were presented. Results show that valid grasps are generated for everyday objects to be used in real time by the robotic dexterous hand.

Future work will address extending the range of objects, and possibly improving the segmentation heuristics, so that a more rich knowledge base can be built, but without compromising the short processing time required for real-time robotic implementation. This will allow a more thorough comparison with existing but more time consuming approaches to grasp synthesis. With respect to the learning, we intend to go beyond the use of histogram distributions. By fitting an ensemble of parametric distributions to the data, a selection model using Bayesian Information Criterion can assign a score to each distribution to select the best distribution that models the data. The final stage of the grasp synthesis can also be improved to better avoid collisions in the execution.

## Acknowledgements

The authors would like to thank Dr. Guillaume Walck from the Institut des Systmes Intelligents et de Robotique, UPMC-Paris, for his work on the integration of all Modules into the General Software and for the tests realized in the robotic platform. We also would like to thank all consortium of HANDLE project for the discussions and cooperative work developed during the progress of the project. The research leading to these results has been partially supported by the HANDLE project, which has received funding from the European Community's 7th Framework Programme under grant agreement ICT 231640; by the Portuguese Foundation for Science and Technology (FCT), the Robotics Institute at Khalifa University Abu Dhabi-UAE, and the Institute of Systems and Robotics, University of Coimbra, ISR-UC.

## References

[1] Grasp project: emergence of cognitive grasping through introspection, emulation and surprise, http://grasp.xief.net/.
[2] Shadow robot dextrous hand, www.shadowrobot.com.

[3] J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-driven grasp synthesis — a survey, IEEE Trans. Robot. (2013).
[4] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, IEEE Trans. Syst. Man Cybern. B 37 (2) (2007) 286–298.
[5] J.J. Steil, F. Röthling, R. Haschke, H. Ritter, Situated robot learning for multi-modal instruction and imitation of grasping, Robot. Auton. Syst. 47 (2004) 129–141.
[6] E. Oztop, M.A. Arbib, Schema design and implementation of the grasp-related mirror neuron system, Biol. Cybernet. 87 (2) (2002) 116–140.
[7] M. Fischer, P.V. der Smagt, G. Hirzinger, Learning techniques in a dataglove based telemanipulation system for the dlr hand, in: IEEE International Conference on Robotics and Automation, 1998.
[8] S. Ekvall, D. Kragic, Interactive grasp learning based on human demonstration, in: IEEE/RSJ International Conference on Robotics and Automation, 2004.
[9] J. Aleotti, S. Caselli, Programming task-oriented grasps by demonstration in virtual reality, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation, 2008.
[10] M. Hueser, T. Baier, J. Zhang, Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration, in: IEEE International Conference on Robotics and Automation, 2006.
[11] J. Romero, H. Kjellstrm, D. Kragic, Human-to-robot mapping of grasps, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation, 2008.
[12] A. Bierbaum, K. Welke, D. Burger, T. Asfour, R. Dillmann, A framework for visually guided haptic exploration with five finger hands, in: Proceedings of the Robotics: Science & Systems, Manipulation Workshop — Sensing and Adapting to the Real World, 2007.
[13] K. Muelling, J. Kober, O. Kroemer, J. Peters, Learning to select and generalize striking movements in robot table tennis, Int. J. Robot. Res. 32 (2013) 263–279.
[14] V. Kruger, D. Herzog, S. Baby, A. Ude, D. Kragic, Learning actions from observations, IEEE Robot. Autom. Mag. 17 (2) (2010) 30–43.
[15] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, Learning object affordances: from sensory motor maps to imitation, IEEE Trans. Robot. 24 (1) (2008) 15–26.
[16] R. Pelossof, A. Miller, P. Allen, T. Jebara, An svm learning appraoch to robotic grasping, in: IEEE International Conference on Robotics and Automation, 2004.
[17] Y. Li, N. Pollard, A shape matching algorithm for synthesizing humanlike enveloping grasps, in: 5th IEEE-RAS Int. Conf. on Humanoid Robots, 2005, pp. 442–449.
[18] J. Bohg, D. Kragic, Learning grasping points with shape context, Robot. Auton. Syst. 58 (4) (2010) 362–377.
[19] A. Saxena, J. Driemeyer, J. Kearns, A.Y. Ng, Robotic grasping of novel objects, Neural Inf. Process. Syst. 19 (2007) 1209–1216.
[20] M. Stark, M.Z.P. Lies, B. Schiele, Functional object class detection based on learned affordance cues, Comput. Vis. Syst. (2008) 435–444.
[21] A.T. Miller, S. Knoop, H.I. Christensen, P.K. Allen, Automatic grasp planning using shape primitives, in: Proceeding of International Conference on Robotics and Automation, ICRA 2003, 14–19 Sep., Tapei, 2003, pp. 1824–1829.
[22] C. Goldfeder, P.K. Allen, C. Lackner, R. Pelossof, Grasp planning via decomposition trees, in: IEEE Int. Conf. on Robotics and Automation, 2007.
[23] E. Lopez-Damian, D. Sidobre, R. Alami, Grasp planning for non-convex objects, in: 36th International Symposium on Robotics, ISR, 2005.
[24] G. Biegelbauer, M. Vincze, Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation, in: international Conference on Robotics and Automation, ICRA'07, 2007.
[25] S. El-Khoury, A. Sahbani, A new strategy combining empirical and analytical approaches for grasping unknown 3d objects, Robot. Auton. Syst. 58 (5) (2010) 497–507.
[26] A. Sahbani, S. El-Khoury, A hybrid approach for grasping 3d objects, in: IEEE/RSJ Int. Conf. intelligent robots and Systems, St. Louis, USA, 2009.
[27] T.A.P. Azad, R. Dillmann, Stereo-based 6d object localization for grasping with humanoid robot, in: In IEEE International Conference on Intelligent Robots and Systems, IROS 2007, San Diego, USA, 2007.
[28] N.B.S. Rodrguez-Jiménez, M. Abderrahim, 3d object reconstruction with a single rgb-depth image, in: International Conference on Computer Vision Theory and Applications, VISAPP, Barcelona, Spain, 2013.
[29] U. Prieur, V. Perdereau, A. Bernardino, Modeling and planning high-level in-hand manipulation actions from human knowledge and active learning from demonstration, in: IEEE International Conference on Intelligent Robots and Systems, 2012.
[30] J.A.C. Ramon, F.T. Medina, V. Perdereau, Finger readjustment algorithm for object manipulation based on tactile information, Int. J. Adv. Robot. Syst. (2013).
[31] C. Rosales, R. Suarez, M. Gabiccini, A. Bicchi, On the synthesis of feasible and prehensile robotic grasps, in: IEEE International Conference on Robotics and Automation, 2012.
[32] M. Gabiccini, A. Bicchi, D. Prattichizzo, M. Malvezzi, On the role of hand synergies in the optimal choice of grasping forces, Auton. Robots 31 (2011) 235–252.
[33] L. Chen, N. Georganas, An efficient and robust algorithm for 3d mesh segmentation, Multimedia Tools Appl. 29 (2) (2006) 109–125.
[34] I. Biederman, Recognition-by-components: a theory of human image understanding, Psychol. Rev. 94 (1987) 115–147.
[35] D.R. Faria, R. Martins, J. Lobo, J. Dias, Probabilistic representation of 3d object shape by in-hand exploration, in: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Taipei, Taiwan, 2010.
[36] D.R. Faria, R. Martins, J. Lobo, J. Dias, Extracting data from human manipulation of objects towards improving autonomous robotic grasping, Robot. Auton. Syst. 60 (3) (2012) 396–410. Special Issue on autonomous grasping.
[37] A.T. Miller, Graspit!: a versatile simulator for robotic grasping, Ph.D. thesis, Ph.D. Thesis, Department of Computer Science, Columbia University, June 2001.
[38] M. Santello, M. Flanders, J.F. Soechting, Postural hand synergies for tool use, J. Neurosci. 18 (1998) 105–115.
[39] D.R. Faria, J. Lobo, J. Dias, Identifying objects from hand configurations during in-hand exploration, in: 2012 IEEE International Conference on Multisensor Fusion and Information Integration, IEEE MFI 2012, Hamburg, September, 2012.
[40] G. Rissanen, Modeling the shortest data description, Automatica 14 (1978) 465–471.
[41] M.R. Gupta, Y. Chen, Theory and use of the em algorithm, Found. Trends Signal Process. 4 (3) (2011) 223–296.
[42] A.H. Barr, Superquadrics and angle preserving transformations, in: IEEE Computer Graphics and Applications, 1981.
[43] F. Solina, A. Leonardis, A. Macerl, A direct part-level segmentation of range images using volumetric models, in: IEEE International Conference on Robotics and Automation, 1994.
[44] L. Chevalier, F. Jaillet, A. Baskurt, Segmentation and superquadric modeling of 3d objects, WSCG 11 (2003) 232–239.
[45] P. Drews Jr., P. Nunez, R. Rocha, M. Campos, J. Dias, Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots, in: Proc. of IEEE ICRA'10, 2010.
[46] A. Jaklic, A. Leonardis, F. Solina, Segmentation and Recovery of Superquadrics, in: Computational Imaging and Vision, vol. 20, Kluwer, Dordrecth, ISBN: 0-7923-6601-8, 2000.
[47] D.R. Faria, R. Martins, J. Lobo, J. Dias, A probabilistic framework to detect suitable grasping regions on objects, in: 10th IFAC Symposium on Robot Control, SYROCO'12, Dubrovinik, Croatia, Sept., 2012.
[48] Handle project — data collection database, 2012. URL: http://paloma.isr.uc.pt/datacollectiondb/handle.
[49] Handle project, 2013 — developmental pathway towards autonomy and dexterity in robot in-hand manipulation, 2013. http://www.handle-project.eu/.
[50] Pcl — point cloud library, 2013. http://pointclouds.org.
[51] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[52] F. Veiga, A. Bernardino, Towards bayesian grasp optimization with wrench space analysis, in: IEEE IROS 2012 Workshop "Beyond Robot Grasping", Vilamoura, Portugal, October, 2012.
[53] M. Ciocarlie, C. Goldfeder, P. Allen, Dimensionality reduction for hand-independent dexterous robotic grasping, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 3270–3275.
[54] N. Hendrich, Deliverable 24: parameterizing and creating new actions, in: HANDLE project, D24, 2012. http://www.handle-project.eu/.

**Diego R. Faria** was born on August 17th, 1979 in Londrina-PR, Brazil. He has been carrying the Ph.D. studies and is researcher at the Institute of Systems and Robotics — Department of Electrical and Computer Engineering — University of Coimbra, Portugal. He earned a Bachelors degree in Informatics Technology (data computing and information) in 2001, and has finished a Computer Science Specialisation in 2002 at the State University of Londrina, Brazil. He holds an M.Sc. degree in Computer Science from the Federal University of Parana, Brazil, since 2005. During the Ph.D. research, Diego Faria was sponsored by a Ph.D. scholarship from the Portuguese Foundation for Technology and Sciences. He has collaborated as researcher on the European Project HANDLE within the 7° framework FP7 (from 2009 to 2013) and on the European project BACS within the 6° framework FP6 (from 2006 to 2008). His research interests are Robotic Grasping, Multimodal Perception, Imitation Learning, Computer Vision and Pattern Recognition.

**Pedro Trindade** has an M.Sc. degree in Electrical and Computer Engineering from the University of Coimbra, Portugal, obtained in 2010. He has been carrying Ph.D. studies and is a researcher at Institute of Systems and Robotics — University of Coimbra, Portugal and Citard Services Ltd, Cyprus. He has recently collaborated on the European Project HANDLE within the EU 7th framework. He is also collaborating on the European project Social Robot within the Seventh Framework Programme FP7, by People Programme, Industry-Academia Partnerships and Pathways (IAPP). His research interests are Intertial Sensing, Sensor Fusion, Computer Vision, and Robotic Grasping.

**Jorge Lobo** (Jorge Nuno de Almeida e Sousa Almada Lobo) was born on the 23rd of September 1971, in Cambridge, UK. In 1995, he completed his five year course in Electrical Engineering at Coimbra University. In April 2002, he received the M.Sc. degree, and in June 2007 he received the Ph.D degree from the University of Coimbra. He was a junior teacher in the Computer Science Department of the Coimbra Polytechnic School, and later joined the Electrical and Computer Engineering Department of the Faculty of Science and Technology at the University of Coimbra, where he currently works as Assistant Professor. He is responsible for courses on Digital Design, Microprocessors and Computer Architecture. His current research is carried out at the Institute of Systems and Robotics, University of Coimbra, working in the field of computer vision, sensor fusion, and mobile robotics. His current research interests focus on inertial sensor data integration in computer vision systems, Bayesian models for multimodal perception of 3D structure and motion, and real-time performance using GPUs and reconfigurable hardware. He has participated in several national and European projects, most recently in BACS, Bayesian Approach to Cognitive Systems, and HANDLE.

**Jorge Dias** was born on March 7, 1960, in Coimbra, Portugal and has a Ph.D. degree on Electrical Engineering at University of Coimbra, specialisation in Control and Instrumentation, November 1994. Jorge Dias conducts his research activities at the Institute of Systems and Robotics (ISR—Instituto de Sistemas e Robótica) at University of Coimbra. Jorge Dias' research area is Computer Vision and Robotics, with activities and contributions on the field since 1984. He has several publications on Scientific Reports, Conferences, Journals and Book Chapters. Jorge Dias teaches several engineering courses at the Electrical Engineering and Computer Science Department, Faculty of Science and Technology, University of Coimbra. He is responsible for courses on Computer Vision, Robotics, Industrial Automation, Microprocessors and Digital Systems. He is also responsible for the supervision of Master and Ph.D. students on the field of Computer Vision and Robotics.