



A New Strategy Combining Empirical and Analytical Approaches for Grasping Unknown 3D Objects

Sahar El-Khoury, Anis Sahbani

► To cite this version:

Sahar El-Khoury, Anis Sahbani. A New Strategy Combining Empirical and Analytical Approaches for Grasping Unknown 3D Objects. *Robotics and Autonomous Systems*, 2010, 58 (5), pp.497-507. 10.1016/j.robot.2010.01.008 . hal-00736018

HAL Id: hal-00736018

<https://hal.science/hal-00736018>

Submitted on 27 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Strategy Combining Empirical and Analytical Approaches For Grasping Unknown 3D Objects

S. El-Khoury, A. Sahbani

Université Pierre et Marie Curie, Paris 6, France

Abstract

This paper proposes a novel strategy for grasping 3D unknown objects in accordance with their corresponding task. We define the handle or the natural grasping component of an object as the part chosen by humans to pick this object with. When humans reach out to grasp an object, it is generally in the aim of accomplishing a task. Thus, the chosen grasp is quite related to the object task. Our approach learns to identify objects handles by imitating humans. In this paper, a new sufficient condition for computing force-closure grasps on the obtained handle is also proposed. Several experiments were conducted to test the ability of the algorithm to generalize to new objects. They also show the adaptability of our strategy to the hand kinematics.

Key words:

Grasp planning, force-closure, learning by imitation

1. Introduction

To operate in everyday environments and assist people, robots have to deal with a wide range of objects. Thus, many robotics applications require manipulation. Grasp synthesis is the central action of objects manipulation and this study will focus on that phase.

The first goal of every grasping strategy is to ensure stability. A grasp is stable if a small disturbance, on the object position or finger force, generates a restoring wrench that tends to bring the system back to its original configuration [14]. Nguyen [43] introduces an algorithm for constructing stable

Email addresses: `s.khoury@lisif.jussieu.fr` (S. El-Khoury),
`anis.sahbani@upmc.fr` (A. Sahbani)

grasps. Nguyen also proves that all 3D force-closure grasps can be made stable. A grasp is force-closure when the fingers can apply appropriate forces on the object to produce wrenches in any direction [9]. Obviously, stability is a necessary but not a sufficient condition for a grasping strategy. When we reach out to grasp an object, we have a goal in our mind or a task to accomplish. Computing task-oriented grasps is consequently crucial for a grasping strategy. Finally, because of the variety of objects shapes and sizes, grasping novel objects is required. This paper proposes a grasp synthesis strategy that meets these constraints.

2. Related Work

A grasping strategy should ensure stability, task compatibility and adaptability to novel objects. In other terms, it should always have an answer to the following question: where to grasp a novel object in order to accomplish a task? Analytical and empirical approaches answer this question differently.

2.1. Analytical Approaches

Analytical Approaches consider the laws of physics, kinematics and dynamics in determining grasps. Such approaches address the problem of computing force-closure and task-oriented grasps. Many works have been developed to compute force-closure grasps. Grasp synthesis approaches dealing with polyhedral objects reduce the force-closure condition to a test of the angles between the faces normals [43] or use the linear model to derive analytical formulation for grasps characterization [47, 18, 29]. Some general approaches, where no restrictions are placed on the object model, were also used [15, 30]. These methods find contact points on a 3D object surface that ensure force-closure. But what about computing good force-closure grasps? For this purpose, different quality criteria were introduced to the grasping literature for computing optimal force-closure grasps achieving the most desirable performance in resisting external wrench loads [41, 23]. These approaches are tackled between optimizing and heuristical techniques. The former compute optimal force-closure grasps by optimizing an objective function according to a pre-defined grasp quality criterion [41]. When objects are modelled with a set of vertices, they search all their combinations to find the optimal grasp [41, 23, 54]. Heuristic methods generate first many grasps candidates, filtered them with a simple heuristic and then choose the best candidate [32, 26, 40]. These approaches find stable grasps adapted for pick

and place operations but are not task-oriented.

Only few works [17, 46, 27, 48] take the task into account. The task wrench space (TWS) models wrenches applied on the grasped object in order to perform a task. Given an object and a task to be executed, Li and Sastry proposed to represent the TWS as a six-dimensional ellipsoid [17]. The latter conforms well the task but its difficult to obtain. The authors were conducted to pre-compute the trajectory followed by the object to accomplish the task. Obviously, this approach is not adapted to new tasks nor to new objects, the whole computation procedure will be repeated. Pollard models the TWS with a six-dimensional unit sphere [46]. Thus, it is assumed that the probability for every wrench direction to occur is equal. This representation has no physical interpretation since wrenches occurring at an object boundary are not uniform. Consequently, the TWS is not uniform as well. Borst approximates the OWS with an ellipsoid in order to model the TWS [27]. This representation takes into account the object geometry and the wrenches it may encounter. But since this representation accounts for different wrenches on the whole object boundary, it does not consider task specific information. Thus, the computed grasp is not the best adapted to a specific task. Haschke [37] optimizes the maximal applicable wrench for a given task wrench direction. However, the paper does not include any information about the corresponding task wrench direction computation. Thus, task-oriented analytical methods suffer from two main problems: the difficulty of modeling a task and the computational effort to find a grasp suitable for the corresponding task. While the selection of task-oriented optimal grasp is very easy for a human hand, it is still a complicated process for a robot hand.

2.2. Empirical Approaches

Empirical grasping methods avoid the computational complexity of analytical techniques by attempting to mimic human grasping strategies. Empirical strategies for grasp planning can be divided into two main kinds: (1) systems based on the observation of the object to be grasped and (2) systems based on the observation of a human performing the grasp. The former techniques generally learn to associate objects characteristics with a hand preshape, while in the latter, a robot observes a human operator performing a grasp and try then to imitate the same grasp. This technique is called in the literature learning by demonstration approach. Different Learning-by-Demonstration frameworks, where the robot observes the human performing

a task and is afterwards able to perform the task itself were proposed in the literature. One of the problems arising in human based learning settings is the one of measuring human performance. Some researchers use data-gloves, map human hand to artificial hand workspace and learn the different joint angles [35, 31], hand preshapes [38] or the corresponding task wrench space [24] in order to perform a grasp. Others use stereoscopy to track the demonstrator’s hand performing a grasp [36] or try to recognize its hand shape from a database of grasp images [49]. Mirror neurones that fire not only when grasping but also when observing an action were also introduced to the grasping problem [19]. These approaches enable objects telemanipulation or grasp type recognition. However, their learning data is based on the hand observation, i.e the joint angles, the hand trajectory or the hand shape. Thus the learning algorithm do not take into consideration the manipulated object properties. Consequently, these methods are not adapted to grasping previously unknown objects. The authors in [19] roughly estimate the size and location of the object and relate them to the hand properties. Kyota [38] finds cylinder-likeness surfaces on the object and associate these surfaces with different hand shapes. They do not take into account object usage. Thus, these approaches can find stable grasps for pick and place operations but are unable to determine a suitable grasp for object manipulation. Grasping strategies based on the object observation analyze its properties and learn to associate them with different grasps. Pelossof’s strategy can predict the quality of a grasp according to a stability criterion [45]. Saxena’s approach [22] find grasping points on mugs handles or on elongated objects mid-points. Such contact points are adapted to some objects in terms of task-compatibility but when this approach encounter elongated objects such as screw-drivers or bottles, it will also identify a grasping region situated at these objects middles. Such grasps are not necessarily adapted to such kinds of objects. Stark’s grasping strategy [52] can only distinguish between two objects classes: handle-graspable (adapted for mugs) and side-graspable (adapted for bottles). This method does not take into account the variety of objects shapes and thus the variety of possible grasps. Li and Pollard strategy [16] determine for one object different grasps and fail to choose the one adapted to the task-requirements. These systems can generalize to new objects but they find either stable grasps or generate for one object different grasps and fail to select automatically the one that best suits the task. When trying to do this autonomously, they encounter the same problem of analytical task-oriented methods, which is task modelling.

2.3. Contribution

Fully autonomous grasping of a previously unknown object remains a challenging problem. In other words, a strategy that learns to associate a grasp to an unknown object/task is still unsolved. We have shown that neither analytical nor empirical approaches can fulfill by themselves the constraints of stability, task compatibility and adaptability to new objects. We propose, in this paper, an approach combining empirical and analytical methods to find appropriate grasps for novel objects. Our strategy associates to each object a handle by imitating humans. This permits to find, for an unknown object, a grasp in accordance with its corresponding task. A new sufficient condition for computing force-closure grasps on the obtained handle is also proposed. This method aims at reducing force-closure grasps computation time.

3. The proposed approach

It's obvious that if we are able to recognize objects, we will also be able to associate a grasp to each object category. Because of the variety of objects shapes and sizes, predicting every possible object the robot could encounter is impossible. Thus, a robot will certainly have to grasp non-identified objects and so are humans. Grasping novel objects is a task humans perform with a great dexterity. To acquire this ability, robots should handle objects in the same manner as humans. Thus, what should the grasping algorithm learn in order to pick a new object in the same manner as humans? In other words, what parameters are relevant to new objects grasping? Are these parameters related to the hand characteristics? Are they related to the object features?

3.1. Grasping By Components: The Concept

According to the Recognition By Components theory of Biederman [10], or RBC, humans are able to recognize objects by separating them into geons, or geometric ions. Geons are composed of different shapes (i.e. cylinders, cones, etc.) that can be assembled in various ways to form an unlimited amount of objects. Biederman suggests that segmenting objects for their identification does not depend on our familiarity with these objects. Thus, we conduct the same process for any object, whether it is familiar or unfamiliar. Consider for example the object shown in Figure 1. Despite its unfamiliarity, we may be able to identify this object by segmenting it into parts at regions of deep concavity. But what about grasping an unfamiliar

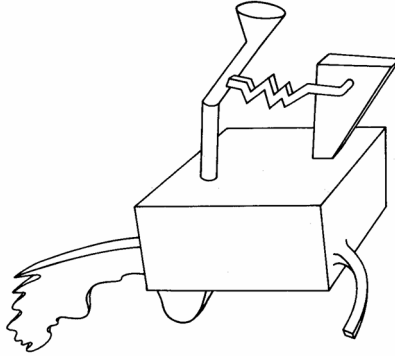


Figure 1: Biederman unfamiliar object [10].

object? Does its part decomposition emphasize a specific grasp?

Assumptions:

When considering objects we use for everyday tasks on a part-representation level, we can make the following assumptions:

- *Objects handles*: Objects are equipped with a part designed specifically to make their grasp easier. Figure 2 shows some familiar objects. The black part indicates the component that humans choose to grasp these objects. Thus, it is also the part that satisfies the task requirements. This part is what we call the object natural grasping component or more simply the object handle. We all agree that the handle of a cup



Figure 2: Some objects used for everyday tasks. The black part indicates the object handle.

or a mug is its curved part and that the handle of a bottle, a pencil or a spoon is their elongated parts. But what about the handle of an unknown object? Which part of the object is there to facilitate its grasp? If we can determine the handle of an unknown object, we can

easily find a grasp of that object adapted to the task it is designated to.

- *Grasping Similarities:* Many objects with similar components are grasped in the same manner. Bags, buckets, mugs and cups are roughly composed of a cylinder and a curved cylinder. Even though the arrangement of these components is different for these objects, they are all grasped by their curved component (Fig. 3). Thus, the choice of an object graspable part is influenced by the shape of its constituting single parts. Objects parts orientation is less relevant to that choice.

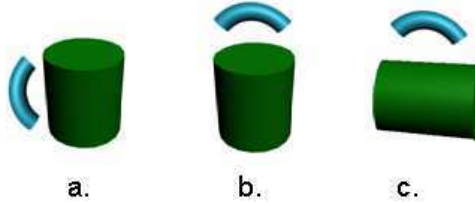


Figure 3: The choice of an object graspable part is influenced by the shape of its constituting parts, independently from their orientations, i.e: a) a mug, b) a bucket and c) a bag are all grasped by their curved part.

- *Handle Vs. Relative Sizes:* The relative sizes of object components is crucial for the graspable part selection. Let us examine some alcohol glasses shapes and sizes. More precisely, we consider wine, champagne and brandy glasses. Although, all these glasses are composed of three parts: the bowl, the stem and the foot, they are grasped differently (Fig. 4). Wine glasses are characterized by their wide bowl which gives the wine the chance to breathe. Champagne flutes are characterized by a narrow bowl on the top. This is designed to keep sparkling wine desirable during its consumption. Wine and champagne glasses are designed to be held by the stem to help prevent the heat from the hand from warming the alcohol. On the other hand, brandy glasses have a short stem. They are designed to be held by the bowl. The wide bowl of the brandy glass accommodates the hand, which warms the brandy for drinking. This example shows that objects are designed

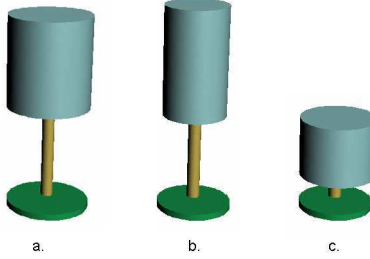


Figure 4: Roughly approximation of: a) a wine glass, b) a champagne flute and c) a brandy glass.

in a way making their grasp easier and that the choice of the graspable component is influenced by the objects parts relative sizes.

In summary, we can say that information about an unknown object parts shapes and sizes may emphasize a specific part for grasping. This leads to the "Grasping By Components" or GBC strategy.

3.2. Grasping By Components: The Strategy

The diagram below (Fig. 5) illustrates a description of the proposed grasping strategy [34]. By representing objects as a set of components, we may identify the graspable one. Hence, our approach is composed of four steps:

Part Segmentation: Objects are first decomposed into single parts at regions of deep concavities. This conforms well with human intuition about parts. A common 3D objects segmentation technique is to compute surface features which contrast boundary and non-boundary points and then to decompose the object into parts at boundary points. The key issue here is how to reliably locate the part boundaries?

Shape Approximation: Since information about objects components shapes and sizes is required, the approximation step ensures a geometrical description of these parts. Therefore, which model has to be used?

Learning: The learning step permits to learn the grasping component of an object using information about its sub-parts shapes and sizes. This step is performed by imitating humans choice of objects graspable parts. The

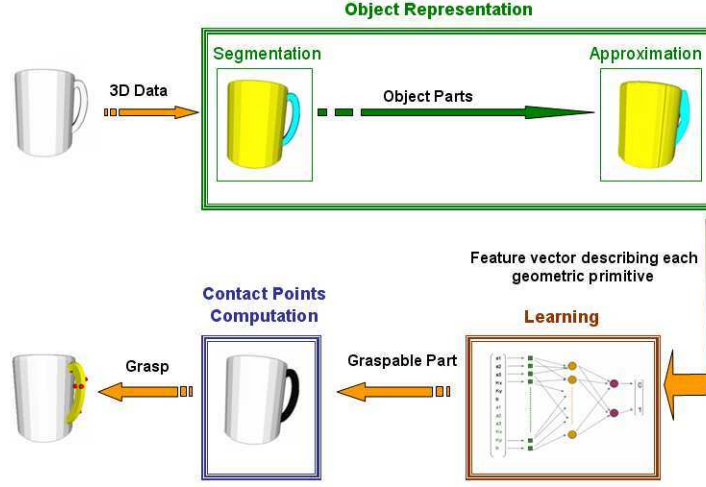


Figure 5: The different steps of the proposed approach.

difficulty here is to determine the training data. What objects should be selected for training and is the algorithm capable to generalize to novel objects?

Contact Points: Once the handle of an object is identified, a grasp is computed on that part. How the contact points are determined? Is the grasp obtained stable?

The following section will focus on object modelling that includes object part segmentation and volumetric part models choice and recovery. The learning step will be detailed in section 5. Computing contact points will be addressed in section 6. The experimental results are detailed in section 7. Section 8 concludes.

4. Object Modeling

Object modeling is obtained by segmenting a 3D object model into its meaningful parts and by describing the shape of each part. We use in our experiments synthetic (CAO models) and real objects (obtained from a 3D laser scanner or from a 3D reconstruction using a vision system) models.

4.1. Segmentation

A segmentation algorithm based on the Gaussian curvature and the concaveness estimation is used [12].

Gaussian curvature and Surface behavior:. Given a vertex p on a polygonal mesh (Fig. 6), the discrete gaussian curvature $k(p)$ is defined by [13]:

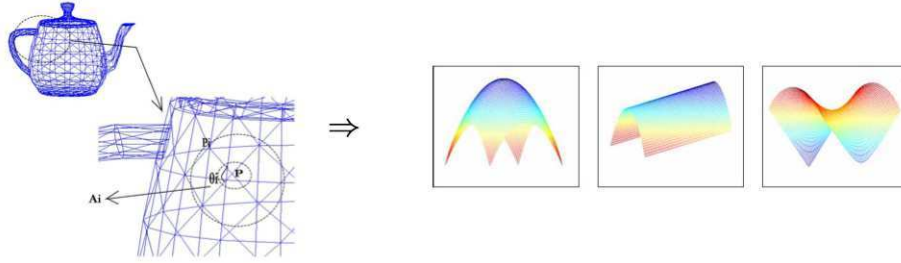


Figure 6: Discrete Gaussian curvature computation. The first, the second, and the third patches show elliptic, parabolic and hyperbolic behavior. Thus they have positive, zero and negative Gaussian curvature respectively.

$$k(p) = \frac{3(2\pi - \sum_i^N \theta_i)}{\sum_i^N A_i} \quad (1)$$

where N is the number of triangles at p ; θ_i , represents the interior angle of the triangle at p ; A_i , represents the area of the corresponding triangle. Thus, a point p belongs to the following categories: elliptic if $k > 0$; hyperbolic if $k < 0$; parabolic if $k = 0$ (Fig. 6). Therefore, we can divide the surface into disjunctive regions by detecting the boundaries with hyperbolic behavior.

Concaveness estimation:. Concave vertices are also pertinent to objects decomposition into meaningful parts. Concaveness and convexity of the vertices on a mesh are estimated with the following equations [50]:

$$N_p = \frac{(\sum_i^N n_i A_i)}{\sum_i^N A_i} \quad p_c = \frac{(\sum_i^N p_i A_i)}{\sum_i^N A_i} \quad (2)$$

$$d = |\overrightarrow{pp_c} \cdot N_p| \quad d_s = \overrightarrow{pp_c} \cdot N_p \quad (3)$$

Where the average plane H of a vertex p is defined by the normal vector N_p of p and a center point p_c , and the distance d from p to H ; A_i , p_i and n_i are the area, the vertex and the normal of the adjacent face around p , respectively, and $\overrightarrow{p_c p}$ denotes the vector from p to p_c ; d_s the signed distance from p to H . Based on the signed distance, the vertex p is defined as convex if $d_s \leq 0$, and as concave if $d_s > 0$.

Gaussian curvature can identify elliptic and hyperbolic behaviors of a 3D polygonal mesh. However, it cannot detect if a corner vertex is concave or convex. Thus the concaveness detection complement the Gaussian curvature. After the estimation of these features on each vertex of the 3D triangular mesh of the object model, vertices are labelled as boundary or inner region vertices. A vertex p is a boundary vertex if it has an hyperbolic behavior or is concave. The segmentation approach is then performed in three steps, minima detection, plateau erosion and region merging. The minima detection step finds the local minima (the non-boundary regions) and mark each minimum with a unique label. The rest areas are considered plateaus. Plateaus are then eroded to their neighbor minima. Finally a region merging step merges the less important regions according to a size criterion to their neighbor regions. More details on these steps can be found in [12]. Since this method uses multi-ring neighborhood in order to compute a 3D object surface features, it succeeds in decomposing low resolution as well as high resolution 3D laser scanned objects (Fig. 7).

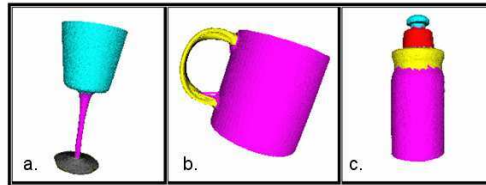


Figure 7: Segmentation of real 3D laser scanned objects.

4.2. Approximation

Object segmentation produces a set of parts. The next task is to generate a description for each one. Each part is represented by a superquadric. With

only few parameters, superquadrics can represent a large variety of standard geometric solids as well as smooth shapes. In order to have a manageable number of superquadrics shapes, we have chosen 7 representative models that span the space of superellipsoids: box, cylinder, sphere, bent box, bent cylinder, tapered box and tapered cylinder [34]. A superquadric surface model is defined by the following implicit equation:

$$f(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \quad (4)$$

Where, a_1 , a_2 and a_3 , define the superquadric size; ϵ_1 and ϵ_2 , determine the shape curvatures that define a smoothly changing family of shapes from rounded to square. This compact model of superquadrics, defined by only five parameters, can model a large set of building blocks like spheres, cylinders and boxes (Fig. 8). In order to increase the flexibility of the model (4), two

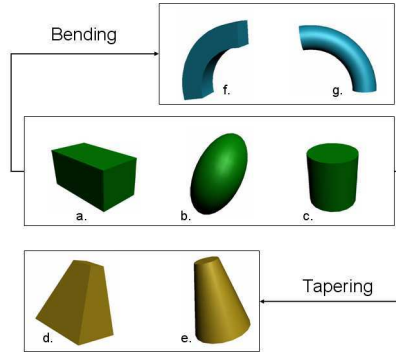


Figure 8: Simple and deformed superquadrics.

deformations, tapering and bending (Fig. 8), are added. Tapering is defined with two parameters k_x and k_y . Bending is defined with the bending angle γ .

5. Learning the Natural Grasping Component

The previous section detailed an object representation as a set of superquadrics. Thus, the remaining issue is obviously to develop a learning algorithm to select the graspable one. This is difficult since objects are composed of a variable number of parts which are of different shapes and sizes.

The question that arise is: What information about the objects is relevant to their grasping? Can this information be reused to grasp novel objects? What could possibly be the training data?

5.1. Object Coding

We previously showed that the choice of the graspable part is influenced by the object components shapes and relative sizes. However, this choice is less influenced by the object parts assembly (section 3.1). The shape and the size of a superquadric are represented by 8 parameters; $a_1, a_2, a_3, \epsilon_1, \epsilon_2, k_x, k_y$ and γ . Thus, a $8 \times S$ column vector V , where S is the object part number, represents the whole object. This object representation is invariant to object translation and rotation. For a scale factor invariance, the size parameters of the object components are represented as the ratio of their most important value.

5.2. Training Data

The proposed learning algorithm should use object components shapes and sizes in order to select the grasping part. We showed that an object sub-parts assembly is less relevant than their shapes and sizes to this choice. Thus, we can consider multi-part objects grasping as an extension of two-part objects grasping knowledge. This leads to a training data constituted of two-part objects. In learning algorithms, a large number of training examples is needed in order to have a good generalization. Collecting real world data is cumbersome. Generating synthetic data is easier and less-time consuming. Therefore, we use synthetic 3D objects models available on Princeton Benchmark [51] and NTU 3D Model Benchmark [11] along with labels indicating the grasping component. Since the learning algorithm should perform an analogue of humans choice of the grasping component, several subjects were asked to identify the grasping part of the corresponding objects.

As for the choice of the two-parts training objects, supervised learning requires a set of objects that can potentially span the space of two superquadrics assembly. Therefore, the choice of the training objects should effectively sub-sample this space. Seven superquadrics are used to model our objects. Thus, the training objects components are chosen to span these 7 superquadrics shapes with different sizes. We use 12 objects for the training set (Fig. 9). Figure (10) shows the steps for generating the training data. It shows first the initial object, its decomposition into single parts, the approximation of each part with a superquadric and finally its corresponding grasping part

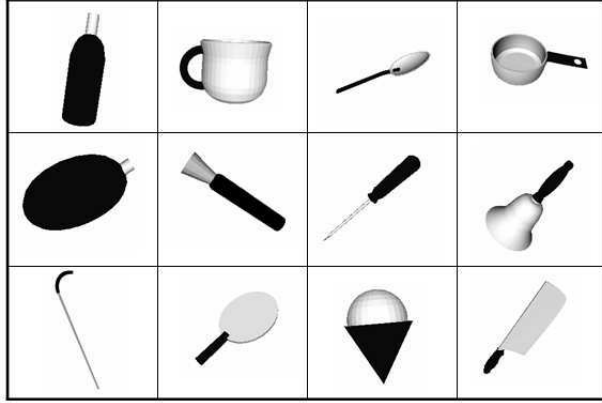


Figure 9: Training objects set. The black part indicates the chosen object handle.

according to humans choice. Additionally, to increase the diversity in our data, once a synthetic model of the object has been created, we vary some properties of the object components such as the size, the bending angle or the tapering parameters without changing the whole appearance of the object. By varying these properties, we generate 72 examples of each object. These examples are divided into training data and testing data.

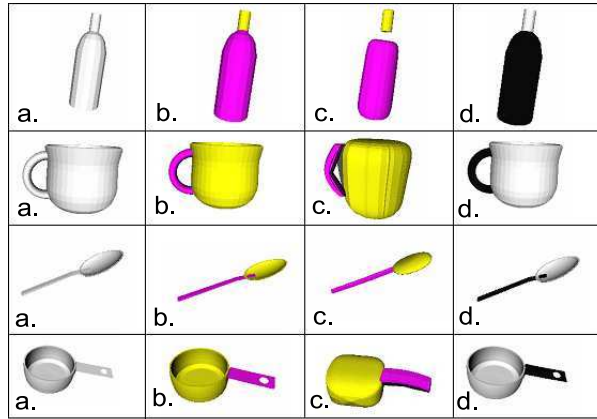


Figure 10: Some two-part objects used for generating the training set. (a) shows the initial 3D object. (b) presents its segmentation into single parts. (c) shows the superquadric approximation of each constituting part. (d) shows the natural grasping part in black.

5.3. Learning Algorithm

A multi-layer perceptron, with one hidden layer, is trained with a typical backpropagation learning algorithm [1] in order to select the grasping part of a two-component object. Eight parameters are sufficient to represent a component shape and size. In the sequel, the first layer has sixteen inputs (training performed on two-parts objects). On the other hand, the output layer represents whether the first or the second component of the object is chosen as grasping part. Thus, the output is a one unit layer. As for the hidden layer, 5 units were chosen empirically.

For multi-part objects, the decision of the grasping component is taken by considering the object parts two by two in a randomly order. In other words, the algorithm starts by choosing a grasping component between two parts of the object. The chosen part is then compared with another component and so on until finding the handle of the multi-part object.

6. From the Grasping Component to the Grasping Points

This section concerns the generation of stable grasps on the object graspable part. The stability of a grasp is characterized by force-closure property. In the past few years, several force-closure tests were proposed [18, 23]. These methods require considerable computation time. Researchers used heuristic approaches to improve performance by randomly generating grasps and filtering them [26] or by generating grasps in respect of specific rules which conducts to necessary but not sufficient condition of force-closure [44]. Our heuristic is original in the sense that it permits not only fast computation but also good quality force-closure grasps generation. This is confirmed by comparing it to the classical convex-hull method [39].

6.1. Force-Closure Preliminaries

This paragraph presents definitions and theorems necessary for our force-closure test elaboration.

Definition 1: According to the definition of Salisbury and Roth [21], a grasp is force-closure if and only if any external wrench can be balanced by the wrenches at the fingertips.

Proposition 1: Salisbury and Roth [21] have also showed that a necessary and sufficient condition for force-closure is that the primitive contact

wrenches resulted by contact forces at the contact points positively span the entire 6-dimensional wrench space. This condition is equivalent to that the origin of the wrench space lies strictly inside the convex hull of the primitive contact wrenches [4, 42].

Proposition 2: For any n -dimensional Euclidean space E^n , $n + 1$ vectors are necessary to positively span E^n [3].

Proposition 3: A set of $n + 1$ vectors v_1, v_2, \dots, v_{n+1} in R^n positively span E^n if and only if v_{n+1} is a unique linear combination of v_i , $i = 1, \dots, n$ and all coefficients are strictly negative [53]. In other words, the $n + 1$ vectors positively span E^n if and only if v_1, v_2, \dots, v_n are linearly independent, thus constitute a basis of E^n , and v_{n+1} is written as:

$$v_{n+1} = \sum_{i=1}^n \alpha_i v_i, \quad \alpha_i < 0$$

6.2. A new sufficient condition for n -finger force-closure grasps

In order to ensure force-closure or determine grasp wrenches that positively span the entire 6-dimensional wrench space, one needs to find: (1) primitive wrenches that constitute a 6D basis and (2) a primitive wrench that can be expressed as a negative linear combination of that basis. But, in which case wrenches associated to hard contact points may form a basis of the wrench space? May a representation in the 3D space of 6D wrenches facilitate the problem? Plücker coordinates represents a 6D contact wrench by the line of action of its corresponding force [5, 28] and Grassmann algebra studies the rank of such lines [2]. We use these two studies to prove the following propositions [33]:

Proposition 4: The 6 lines on the sides of a tetrahedron are independent, and thus form a basis of R^6 (Fig. 11).

Proposition 5: Wrenches associated to 3 non-aligned contact points are of rank 6.

Proposition 6: Assume that the grasp of $n - 1$ non-aligned fingers is not force-closure. Suppose that $\{b_i\}_{i=1..k}$ is the k -dimensional (where $k = 6$) basis

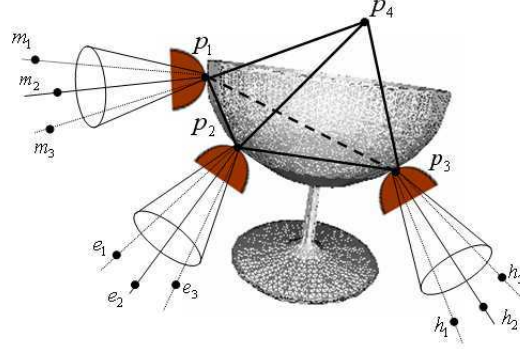


Figure 11: The wrenches of rank 3 associated to the frictional contact points p_1 , p_2 and p_3 .

associated to their corresponding contact wrenches. A sufficient condition for a n -finger force-closure grasp is that there exists a contact wrench γ such that:

- γ is inside the linearized friction cone of the n th finger
- (5)

- $\gamma = \sum_{i=1}^k \beta_i b_i, \beta_i < 0$

$$\Rightarrow \gamma = B\beta \Rightarrow \beta = B^{-1}\gamma$$
(6)

where $B = [b_1, b_2, \dots, b_k]$ is a $k \times k$ matrix and $\beta = [\beta_1, \beta_2, \dots, \beta_k]^T$ is a $k \times 1$ strictly negative vector. Thus, a simple multiplication by B^{-1} permits to test if a contact wrench γ , and consequently the location of the n th contact point, ensures a force-closure grasp.

Proof. A necessary and sufficient condition for force-closure is that the primitive contact wrenches resulted by contact forces at the contact points positively span the entire k -dimensional wrench space [21]. A set of $k + 1$ vectors in R^k positively span E^k if and only if the $(k + 1)$ th vector is a unique linear combination of the other k vectors and all coefficients are strictly negative, (from *proposition 2*). The $k + 1$ vectors $\{\gamma, b_1, b_2, \dots, b_k\}$ satisfy these conditions and thus positively span R^k . ■

6.3. Quality criterion of the $n - 1$ fingers locations

At this point, we showed (*proposition 6*) that to achieve force-closure, we generate randomly locations of $n - 1$ non-aligned fingers. A position

of the n th finger is chosen such that an associated contact wrench can be uniquely expressed as a strictly negative linear combination of one of the first generated $n - 1$ fingers wrench basis. Our objective is to ensure fast robust force-closure grasps generation. In our case, force-closure grasps fast computation and robustness are strongly linked. In order to understand how the two latter are tied together, one should notice that generating a n -finger good grasp will depend on the generation of the first $n - 1$ fingers. A good choice of their locations will induce on one hand robust grasps and on the other hand more locations for the n th finger on the object surface guaranteeing force-closure and consequently fast computation. Thus, we need to find a criterion that quantifies a placement of the $n - 1$ first fingers that induces a good quality $6D$ wrench basis. Let F_f , where $f = 1..(n - 1)$ be the first generated $n - 1$ fingers. The quality criterion proposed for 3D case is the extension of the one in 2D. In a 2D case, a wrench basis is represented by three points in the 3D space that constitute with the wrench space origin a tetrahedron. A wrench that ensures force-closure grasp is a wrench that can be uniquely expressed as a strictly negative linear combination of the 3D basis. Thus, the larger the tetrahedron, the more choices we have for such a wrench. Different tetrahedron quality measures were proposed in the literature especially in the field of mesh optimization. One of the most used quality is $Q = \frac{V}{h_{max}^3}$ [6], where V is the volume of the tetrahedron and h_{max} is its maximal edge length. Dealing with 3D objects grasps involves $6D$ wrenches. Thus, instead of computing 3D tetrahedra volumes, we are conducted to calculate volumes of $6D$ hypertetrahedra. As the volume V of a tetrahedron can be expressed as the product of a constant and the determinant of the tetrahedron vertices, the computation of a 6-volume hypertetrahedron could be viewed as a determinant calculation:

$$Q(\{F_f\}) = \max_{i,i=1..nb} \frac{\beta \cdot \det(w_1^i, w_2^i, w_3^i, w_4^i, w_5^i, w_6^i)}{h_{i \ max}^6} \quad (7)$$

Where $\{w_j^i\}_{j=1..6}$ is a $6D$ wrench basis; nb is the number of $6D$ basis corresponding to the $n - 1$ fingers; $h_{i \ max}$ is the maximal wrench length. This quality measure is used in the experiments to generate the first $n - 1$ contacts locations on 3D objects. With such a criterion, our approach is four times faster than the classic convex-hull method and the force-closure grasps obtained are of better quality. For further details on these results, the reader should refer to [33]. In the following, we present the algorithm taking into account this quality measure for generating robust force-closure grasps.

Proposed Algorithm

Require: - 3D points representing the object
- Linearized friction cone at each point and corresponding wrenches

Ensure: - n fingers force-closure grasp

```
1: L = Rand_Na_Fingers(n-1)
2: ntry  $\leftarrow$  0
3: L_basis = Find_Basis (L_wrenches)
4: q_L = quality(L_basis)
5: if q_L < threshold then
6:   Go to step 1
7: end if
8: vertex = Rand_Finger(1)
9: ntry  $\leftarrow$  ntry+1
10: FC = Force_Closure_Test(vertex, r_basis)
11: if (!FC) and (ntry  $\leq$  nmax ) then
12:   Go to step 8
13: else
14:   or Go to step 1
15: end if
```

Given a representation of an object along with normal directions and a friction coefficient, wrenches associated to each of its vertices are firstly computed. In order to obtain n -finger force-closure grasps, the function *Rand_Na_Fingers* generates randomly, locations of non-aligned $n - 1$ fingers on the object surface. A number *L_basis* of 6-dimensional basis from the wrenches associated to these $n - 1$ contacts are determined by *Find_Basis*. The quality of these basis is computed with *quality* function. If the latter is below a threshold, we proceed at the generation of other $n - 1$ fingers locations. If the quality of at least one of the *L_basis* is above the specified threshold, an object vertex is then randomly chosen by *Rand_Finger* and tested for ensuring a n -finger force-closure grasp with *Force_Closure_Test*. If the n -finger grasp ensures force-closure the algorithm finishes. Otherwise, *ntry* permits to choose between generating novel $n - 1$ fingers locations or testing another object vertex for force-closure with the basis of the same $n - 1$ fingers. Note that the choice of the threshold and *nmax* is crucial for the algorithm force-closure grasps computation time. The threshold value cannot be determined analytically. It varies with the object shape and thus

is chosen empirically.

7. Experimental Results

We proposed a grasping strategy that describes objects as an assembly of parts and then proceeds to the identification of the handle or the Natural Grasping Component (NGC) in accordance with humans choice. This section aims at testing how well learned grasping skills generalize to new objects. Thus, an algorithm is trained to grasp a small set of objects and tested on a much larger set of everyday items. We begin by validating the learning algorithm model. Two generalization experiments were performed. The first one tests the algorithm on objects belonging to the same categories as the training data but of different shapes and sizes. The second one considers objects that are completely different from those of the training set. We use for our tests synthetic 3D objects models available on Princeton Benchmark [51] and NTU 3D Model Benchmark [11]. We also tested the algorithm on real objects models obtained from a 3D laser scanner and from 3D reconstruction using a vision system.

7.1. Validation of the Learning Algorithm Model

A 10-fold cross validation procedure is employed to validate the learning algorithm. Thus, our training data (72 examples for each of the 12 two-parts objects) is divided randomly into 10 parts. In a first step, the first part is taken apart and used for test data while the 9 remaining parts are used for training data. In a second step, the second part is considered as testing data while the remaining 9 parts are considered as training data. This procedure is repeated ten times. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. The 10 results from the folds are then averaged to produce a single score estimation of the training and testing data. We have an average of 99.45% for the training data and of 98.97% for testing data.

7.2. First Generalization Test

We tested the algorithm on multi-part objects belonging to the same categories as the training data but of different shapes and sizes. These objects are such as bottles, spoons, knives, pencils etc. Some of these objects along with

their obtained grasping part are shown in (Fig 12). This figure illustrates objects decomposition as well as their graspable part in black. The motivation behind this experiment is that if our algorithm does not work on multi-part objects similar to the training data, then we must conclude that our feature set is not sufficiently discriminative. We use for this test 17 objects. For such



Figure 12: Some objects belonging to the same categories as the objects of the training data. The first row shows objects decomposition. The second row shows in black the grasping part identified by the algorithm.

objects, the algorithm generalizes very well and was capable of finding each time the handle that human choose to grasp the corresponding object.

7.3. Second Generalization Test

In a second time we tested the algorithm on 54 objects that are completely different from those of the training set. This experiment is useful to test the algorithm ability to generalize to completely novel objects. Several subjects were asked to grasp these objects in order to accomplish a task. We do not specify the task that should be performed. The subjects were supposed to identify objects graspable parts whether they recognize the object or not. Twenty seven objects, *AO (Agreed Objects)*, were grasped by the same manner. On the other hand, the remaining 27 objects, *CO (Confusing Objects)*, induced confusion and the subjects chose different parts to grasp them. We remind the reader that our aim is to imitate humans choice of the graspable part. The distinction between *AO* and *CO* objects is necessary for measuring our algorithm performance. Their success grasp rate is computed differently.

7.3.1. Success Grasp Rate for Agreed Objects

The different subjects totally agreed on the Agreed Objects handles. Thus, for *AO* objects, whenever the algorithm selects for grasping a part

different from the one identified by the subjects, it is considered a failure. Some of the *AO* objects are shown in (Fig. 13) along with the graspable part identified by the algorithm in black. Objects parts that are marked with a cross are the ones corresponding to humans choice. The system succeeds to find the correct graspable parts for 22 *AO* objects, which corresponds to a successful grasp rate of 81%. This rate shows that features such as sizes and shapes of novel objects subparts are about 81% discriminative to determine the object natural grasping part.

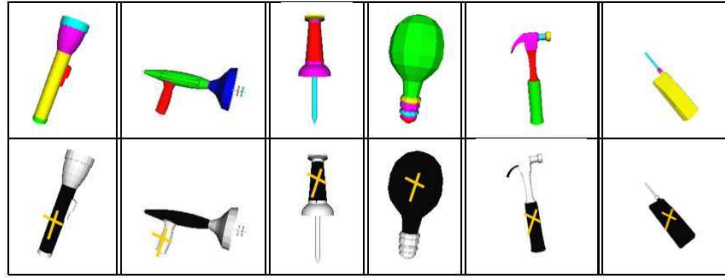


Figure 13: Examples of *AO* objects. The black part indicates the corresponding object graspable part identified by the algorithm. The crossed part indicates the one chosen by humans.

7.3.2. Success Grasp Rate for Confusing Objects

Different parts were chosen by the subjects to grasp Confusing Objects. Since humans grasp these objects in various ways, two successful rate may be computed: a successful grasp may be a grasp that identifies the object part chosen by most people, or a successful grasp may be a grasp that identifies a part chosen by at least one person. Otherwise, failure occurs. Figure (14) shows some examples of *CO* objects. The black part indicates the one chosen by the system and the cross-marked part is the one corresponding to most people choice. The algorithm succeeds to find, for 15 *CO* objects, the part selected by most people. This corresponds to a successful grasp rate of 55%. When considering a grasp rate on the basis of "at least chosen by one person", the algorithm perform well for 23 *CO* objects which corresponds to a rate of 85%.

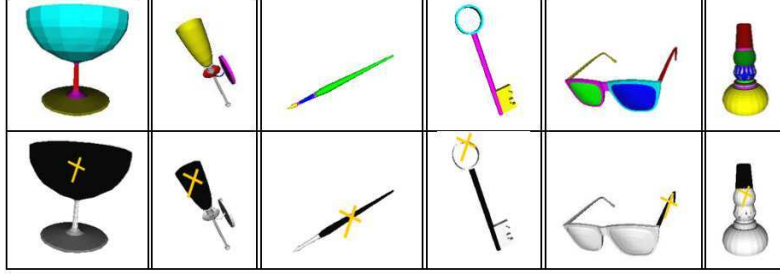


Figure 14: Examples of *CO* objects. The black part indicates the system choice. The cross-marked parts indicate most humans choice.

7.4. Grasping By Components

We proposed a grasping strategy that determines in a first place an unknown object handle and is able then to generate, on this component, contact points ensuring the grasp stability. This section shows in a first place examples of contact points computation on the objects graspable parts. It shows then the adaptability of our strategy to the hand kinematics.

7.4.1. Grasping Strategy Computation Time

Tables 1 and 2 show the computation time corresponding to the generation of a four-finger force-closure grasp on synthetic and real 3D objects models with different resolutions. This time includes the different steps of the grasping strategy; segmentation, approximation, grasping part selection and force-closure grasp computation. Notice that the computation time increases with the number of vertices constituting the object model.

Table 1: Generating 4-finger force-closure grasps for synthetic objects.

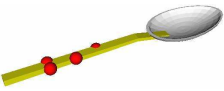



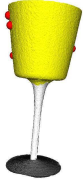

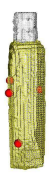
Objects Grasps			
Vertices Number	629	1725	7613
GBC time(s)	2.59	3.73	7.82

Table 2: Generating 4-finger force-closure grasps for laser scanned objects and an object model obtained using vision (the last column).

Objects Grasps				
Vertices Number	183534	137421	122307	7360
GBC time(s)	267.63	173.8	159.82	8.87

7.4.2. Grasping by Taking into account the Hand Kinematics

At this point, our grasping strategy identifies an unknown object handle and generates contact points on it with the only constraint of stability. Dealing with a robotic hand model induces additional kinematical and geometrical constraints. Taking these constraints into account results in limiting possible locations for the contact points on the graspable part. The latter should be kinematically feasible for the fingers and they should also avoid collision with the hand, the remaining fingers and the object. Consequently, these contacts should be generated in respect of the accessibility domains of the fingers. Furthermore, a grasp involves several closed kinematic loops between the fingers and the object. Randomly generation of a closed kinematic chain is very difficult. In order to handle these closed kinematic chains and inspired by the thesis of J.P. Saut [56], we propose to adapt the RLG (Random Loop Generator) algorithm [55] to our grasping strategy. RLG aims at handling closed kinematic loops by dividing them into active and passive parts. The idea of the algorithm is to reduce the closed kinematic chain complexity iteratively until the active part becomes reachable by all passive chain segments simultaneously. In our case, the object is the active part while the fingers constitute the passive parts. A grasp can occur when the object is reachable by all the fingers. The reachable workspace of a kinematic chain is defined as the volume which the end-effectors can reach. RLG approximates such volume with a sphere. Figure 15 illustrates an example of the reachable workspace of a finger. It also shows the intersection between this space and the object. Thus, the finger should be placed on this intersection. The placement of the first finger is then taken into account when computing

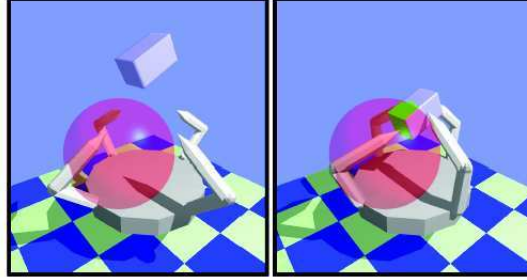


Figure 15: The red sphere represents the reachable workspace approximation of a finger. The green part stands for the object surface reachable by the finger. Thus, this finger should be placed on that green part [56].

the second finger reachable workspace and so on until the placement of all fingers. We modify our grasping strategy to take these constraints into consideration. Figure 16 shows several grasps obtained using DLR and Rutgers hands models and GraspIT simulator. The latter uses PQP algorithm to detect collisions [39].

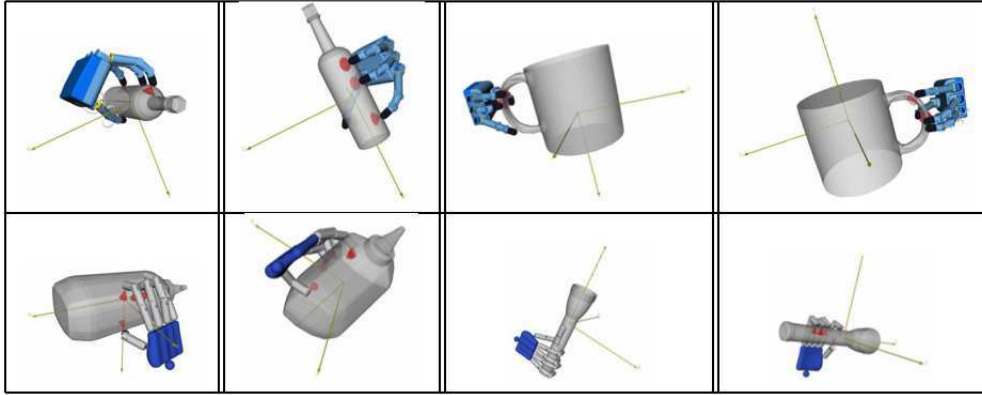


Figure 16: The first row shows generating 4-finger force-closure grasps using DLR hand model in GraspIT simulator. The second row shows examples with the Rutgers hand.

8. Conclusion

This paper addresses the problem of grasping unknown objects. The latter can be stated as follows: given a previously unknown object, determine a

set of contacts on the object surface appropriately in order to ensure stability and to successfully perform a task. Despite the quantity of relevant work in the field, this problem remains challenging and tackles with two main issues:

- Task modelling.
- Generalizing learned grasping skills to new objects.

We overcome these difficulties and propose a strategy that associates a grasp to an unknown object/task by taking inspiration from humans behavior and Biedermann’s theory of Recognition By Components. The latter states that humans recognize objects by representing them as an assembly of parts. On a part-representation level, it seems obvious that each object is equipped with a part designed specifically to make its grasp easier, its handle. We define a handle or the natural grasping component of an object as the part chosen by humans to pick this object with. Our method learns humans choice of the grasping component based on information such as objects sub-parts shapes and sizes. Thus, objects are decomposed into single parts and each part is then fitted with a superquadric. The grasp stability is obtained by determining contact points on the object handle verifying force-closure property.

We implemented the proposed approach and tested its ability to generalize on previously unknown objects models whether synthetic or real via laser scanning or 3D reconstruction using a vision system. The experiments show that the algorithm succeeds in imitating humans when grasping unknown objects. A score of about 80% is obtained. Thus, we can conclude that geometric features such as objects sub-parts shapes and sizes, are indeed relevant to objects grasping. The gap of 20% is due to the fact that some objects have similar shapes but different functionalities, hence, they are grasped differently. Once the object graspable part is determined, we process at the generation of grasping contact points on it. For this purpose, a new sufficient force-closure grasp condition is introduced. It aims at reducing good force-closure grasps computation time. It generates locations of $n-1$ non-aligned fingers according to a quality criterion and finds then locations of the n th finger that ensures force-closure. Finally, we also adapted the algorithm to take into consideration a robotic hand model and corresponding kinematics constraints.

Nowadays, many researchers try to understand humans behavior by programming neuroscience theories in humanoid robots and vice versa [7]. Our work is also based on the psychological RBC theory and other neuroscience studies. But, will the high scores obtained validate the corresponding theories? Are primitives such as objects sub-parts shapes and sizes sufficient to identify novel objects grasps? Do we actually understand how people grasp unknown objects? The work presented in this paper is a first step towards a full understanding of humans behavior and may be our learning algorithm needs to be enhanced to take into account recent/future studies on humans brain.

Acknowledgments

This work is supported by the French ANR (Agence Nationale pour la Recherche) programs under grant No C08527 (ABILIS Project).

References

- [1] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [2] A. Dandurand, *The rigidity of compound spatial grid*, Structural topology 10, 1984.
- [3] A.J. Goldman and A.W. Tucker, *Polyhedral Convex Cones*, Princeton University Press, 1956.
- [4] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical introduction to robotic manipulation*, Orlando, FL: CRC, 1994.
- [5] O. Veblen, J.W. Young, *Projective geometry*, the Athenaeum press, 1910.
- [6] P.D. Zavattieri, G.C. Buscaglia and E.A. Dari, *Finite element mesh optimization in three dimensions*, 1995.
- [7] C.G. Atkeson and J.G. Hale, F.Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar and M. Kawato, *Using Humanoid Robots to Study Human Behavior*, IEEE Intelligent Systems, vol. 15(4), pp. 46-56, 2000.

- [8] A.H. Barr, *Superquadrics and angle-perserving transformations*, IEEE Computer Graphics and Applications, vol. 1, pp. 11-23, 1981.
- [9] A. Bicchi, *On the closure properties of robotic grasping*, International Journal of Robotics Research, vol. 14, no. 4, pp. 319-334, 1995.
- [10] I. Biederman, *Recognition-by-Components: A Theory of Human Image Understanding*, Psychological Review, vol. 94, pp. 115-147, 1987.
- [11] D.Y. Chen and X.P. Tian and Y.T. Shen and M. Ouhyoung, *On Visual Similarity Based 3D Model Retrieval*, Computer Graphics Forum (EUROGRAPHICS'03), vol. 22(3), pp. 223-232, 2003.
- [12] L. Chen and N.D. Georganas, *An efficient and robust algorithm for 3D mesh segmentation*, Multimedia Tools Appl., vol. 29(2), pp. 109-125, 2006.
- [13] A. Gray, *The Gaussian and mean curvatures*, Modern Differential Geometry of Curves and surfaces, pp. 279-285, 1993.
- [14] W.S. Howard and V. Kumar, *On the stability of grasped objects*, IEEE Transactions on Robotics and Automation, vol. 12(6), pp. 904-917, 1996.
- [15] J.W. Li, H. Liu and H.G. Cai, *On Computing Three-finger Force-Closure Grasps of 2D and 3D Objects*, IEEE Transactions on Robotics and Automation, 19(1), 2003.
- [16] Y. Li, J.L. Fu and N. Pollard, *Data-Driven Grasp Synthesis Using Shape Matching and Task-Based Pruning*, IEEE Transactions on Visualization and Computer Graphics, vol. 13(4), pp. 732-747, 2007.
- [17] Z. Li and S.S. Sastry, *Task-Oriented Optimal Grasping by Multifingered Robot Hands*, IEEE Journal of Robotics and Automation, vol. 4(1), 1988.
- [18] Y. H. Liu, *Qualitative test and force optimization of 3-D frictional form closure grasps using linear programming*, IEEE Transactions on Robotics and Automation, vol. 15, no. 1, 1999.
- [19] E. Oztop and M. A. Arbib *Schema Design and Implementation of the Grasp-Related Mirror Neuron System*, Biological Cybernetics 87: (2), pp. 116-140, 2002.

- [20] A.P. Pentland, *Perceptual organization and the representation of natural form*, Artificial Intelligence, vol. 28, pp. 293-331, 1986.
- [21] J.K. Salisbury and B. Roth, *Kinematic and force analysis of articulated hands*, ASME J. Mech., Transmissions, Automat., Design, vol. 105, pp. 33-41, 1982.
- [22] A. Saxena and J. Driemeyer and J. Kearns and A.Y. Ng, *Robotic Grasping of Novel Objects using Vision*, The International Journal of Robotics Research, vol. 27(2), pp. 157-173, 2008.
- [23] X. Zhu and J. Wang, *Synthesis of Force-Closure Grasps on 3D Objects Based on the Q Distance*, IEEE Transactions on robotics and Automation, vol. 19, no. 4, 2003.
- [24] J. Aleotti and S. Caselli, *Programming Task-Oriented Grasps by Demonstration in Virtual Reality*, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation, 2008.
- [25] J. Aleotti and S. Caselli, *Robot Grasp Synthesis from Virtual Demonstration and Topology-Preserving Environment Reconstruction*, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007.
- [26] Ch. Borst, M. Fischer and G. Hirzinger, *Grasping the dice by dicing the grasp*, International Conference on Intelligent Robots and Systems, 2003.
- [27] Ch. Borst, M. Fischer and G. Hirzinger, *Grasp Planning: How to Choose a Suitable Task Wrench Space*, In Proceedings of IEEE International Conference on Robotics and Automation, 2004.
- [28] H. Crapo, *A combinatorial perspective on algebraic geometry*, Colloquio Int. sulle Teorie Combinatorie, Roma, 1973.
- [29] D. Ding, Y.H. Liu, and S. Wang, *Computing 3D optimal form-closure grasps*, Proc. IEEE International Conference on Robotics and Automation, 2000.

- [30] D. Ding, Y.H. Liu, and S. Wang, *On computing immobilizing grasps of 3-D curved objects*, Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, pp. 11-16, 2001.
- [31] S. Ekvall and D. Kragic, *Interactive grasp learning based on human demonstration*, IEEE/RSJ International Conference on Robotics and Automation, New Orleans, USA, 2004.
- [32] M. Fischer and G. Hirzinger, *Fast planning of precision grasps for 3D objects*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 120126, 1997.
- [33] S. El-Khoury and A. Sahbani, *On Computing Robust N-Finger Force-Closure Grasps of 3D Objects*, IEEE International Conference on Robotics and automation, Japan, 2009.
- [34] S. El Khoury, A. Sahbani and V. Perdureau, *Learning the natural grasping component of an unknown object*, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2957-2962, San Diego, CA, USA, 2007.
- [35] M. Fischer, P. Van der Smagt and G. Hirzinger, *Learning techniques in a dataglove based telemanipulation system for the DLR Hand*, Proc. IEEE International Conference on Robotics and Automation, 1998.
- [36] M. Hueser, T. Baier and J. Zhang, *Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration*, Proc. IEEE International Conference on Robotics and Automation, 2006.
- [37] R. Haschke and J.J. Steil and I. Steuwer and H. Ritter, *Task-oriented quality measures for dextrous grasping*, Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, pp. 689-694, 2005.
- [38] F. Kyota, T. Watabe, S. Saito and M. Nakajima, *Detection and Evaluation of Grasping Positions for Autonomous Agents*, In International Conference on Cyberworlds, pp. 453460, 2005.
- [39] A.T. Miller and P.K. Allen, *Examples of 3D grasp quality computations*, Proc. IEEE International Conference on Robotics and Automation, pp. 1240-1246, 1999.

- [40] A.T. Miller, S. Knoop, P.K. Allen and H.I. Christensen, *Automatic grasp planning using shape primitives*, Proc. IEEE International Conference on Robotics and Automation, 2003.
- [41] B. Mirtich and J. Canny, *Easily Computable Optimum Grasps in 2D and 3D*, In Proceedings of IEEE International Conference on Robotics and Automation, vol. 1, pp. 739-747, 1994.
- [42] D.J. Montana, *The condition for contact grasp stability*, in Proc. IEEE International Conference on Robotics and Automation, pp. 412-417, 1991.
- [43] V.D. Nguyen, *Constructing Stable Grasps in 3D*, Proc. IEEE International Conference on Robotics and Automation, pp. 234-239, 1987.
- [44] N. Niparnan and A. Sudsang, *Positive span of force and torque components of four-fingered three-dimensional force-closure grasps*, Proc. IEEE International Conference on Robotics and Automation, 2007.
- [45] R. Pelosof, A. Miller, P. Allen and T. Jebara, *An SVM learning approach to robotic grasping*, In Proceedings of IEEE International Conference on Robotics and Automation, 2004.
- [46] N.S. Pollard, *Parallel algorithms for Synthesis of Whole-Hand Grasps*, In Proceedings of IEEE International Conference on Robotics and Automation, 1997.
- [47] J. Ponce, S. Sullivan, J.D. Boissonnat and J.P. Merlet, *On Characterizing and Computing Three- and Four-Finger Force-Closure Grasps of Polyhedral Objects*, Proc. of Int. Conf. on Robotics and Automation, pp. 821-827, 1993.
- [48] M. Prats, P.J. Sanz and A.P. del Pobil, *Task-Oriented Grasping using Hand Preshapes and Task Frames*, In Proceedings of IEEE International Conference on Robotics and Automation, 2007.
- [49] J. Romero, H. Kjellstrm and D. Kragic, *Human-to-Robot Mapping of Grasps*, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation, 2008.

- [50] W. Schroeder, J. Zarge and W. Lorensen, *Decimation of triangle meshes*, Proc. SIGGRAPH, Computer Graphics, vol. 25(3), pp. 6570,1992.
- [51] P. Shilane and P. Min and M. Kazhdan and T. Funkhouser, *The princeton shape benchmark*, In Proceedings of Shape Modelling International, 2004.
- [52] M. Stark, P. Lies, M. Zillich and B. Schiele, *Functional object class detection based on learned affordance cues*, Computer Vision Systems, pp. 435-444, 2008.
- [53] R. Wagner, Y. Zhuang and K. Goldberg, *Fixturing faceted parts with seven modular struts*, IEEE International Symposium on Assembly and Task Planning, USA, 1995.
- [54] X. Zhu and H. Ding, *Planning Force-Closure Grasps on 3-D Objects*, In Proceedings of IEEE International Conference on Robotics and Automation, 2004.
- [55] J. Cortes, *Motion planning algorithms for general closed-chain mechanisms*, PhD thesis, 2003.
- [56] J.P. Saut, *Planification de mouvement pour la manipulation dextre d'objets rigides*, PhD thesis, 2007.



Sahar El-Khoury has an Electrical Engineering Degree from Ecole Supérieure d'Ingénieurs de Beyrouth (ESIB) in 2004. She received her M.Sc. and PhD degree in Robotics and Intelligent Systems from Université Pierre et Marie Curie (Paris 6) in 2005 and 2008. Her research interests have been focused on grasp planning with a multi-fingered robotic hand.



Anis Sahbani is an Associate Professor at Pierre et Marie Curie University (Paris 6). He holds his research activities at the Intelligent Systems and Robotics Institute (ISIR-CNRS-UMR 7222). He has been principal investigator for several research projects and has been exploring different topics on robotics control, computer vision and in particular manipulation planning. He has an Electrical Engineering Degree from Ecole Nationale d'Ingénieurs de Tunis in 1998. MS in Automation and Computer engineering 1999. He received his PhD in Robotics and Computer Science from University of Paul Sabatier, LAAS-CNRS in 2003.