Handling Objects By Their Handles

Sahar El-Khoury and Anis Sahbani

Abstract— **This paper presents an efficient method to decide robust grasps given new objects using example-based learning. A robust grasp is a stable grasp, suitable for object manipulation. Adaptability to object manipulation is ensured by imitating the human choice of the object grasping component, its handle. Stability is obtained by computing contact points, ensuring force-closure property, on that handle.**

I. INTRODUCTION

Grasping is the central action of object manipulation. In other words, objects should be held appropriately in order to successfully perform a task. This is extremely difficult. Even the most common objects are from different shapes and sizes. Thus, fully autonomous grasping of a previously unknown object remains a challenging problem. However, humans are capable of learning grasps through experience. In this study, we present an approach that makes a robot capable of grasping unknown objects in the same manner as humans when performing everyday tasks.

In general, a successful grasp configuration has to satisfy three main sets of constraints: constraints due to the robotic hand and its fingers capabilities, constraints due to object geometric features and constraints due to task requirements. Several learning techniques have been applied to robotic grasping to meet these constraints. The authors in [8] presents a setup to control a four-finger anthropomorphic robot hand using a dataglove. To be able to accurately use the dataglove a nonlinear learning calibration using a neural network technique has been implemented. Based on the dataglove calibration, a mapping of human and artificial hand workspace can be realized. A similar framework, based on human demonstration, was proposed in [7]. A glove with position sensors gives the location of the fingers and palm. Given a mapping between human hand and different robotic hands, an algorithm is proposed in order to learn the different joint values for the robotic hand. Instead of using a dataglove, a vision and audio based approach was proposed in [12]. The robot stereoscopically tracks the demonstrator's hand in real-time. Then an unsupervised learning method is used in combination to reinforcement learning to let the robot learn the demonstrated grasping trajectories. These approaches enable objects telemanipulation or grasp type recognition. However, their learning data does not take into consideration the manipulated object properties. Thus these methods are not adapted to grasping previously unknown objects.

In a biologically motivated perspective, Oztop and Arbib [19] proposed a very detailed model of the functioning of mirror neurons in grasp learning. They propose a hand-object state association schema that combines the hand related information as well as the object information available. This method is capable of grasp recognition and execution (pinch, precision or power grasp) of simple geometric object models, since the only object features used are the object size and location. Pelossof et al. [20] use Support Vector Machines to learn what is a good grasp for a robotic hand. Their method can approximate the grasp quality for a new set of grasping parameters and select an optimal grasp from the space of grasping parameters of an object. Once more, the authors use simple object representation in their learning algorithm, such as spheres, cylinders etc. Thus, these approaches can find stable grasps for pick and place operations but are unable to determine a suitable grasp for object manipulation.

When a complete $3D$ model of the object is available, Li and Pollard [13] treated grasping as a shape matching problem. Based on the idea that many grasps have similar hand shapes, they construct a database of grasp examples. Thus, given a model of a new object to be grasped, shape features of the object are compared to shape features of hand poses in the database in order to identify candidate grasps. This approach presents many possibilities to grasp an object. To complete a task, the user should manually select the desired one among them. Our approach overcomes this problem by learning the human choice of an object graspable part. Thus, given a new object, our system will identify its handle and consequently find a robust grasp to manipulate it.

II. OUR APPROACH

Castiello [4] showed that cognitive cues and previously learned knowledge both play major roles in visually guided grasping in humans and in monkeys. This indicates that learning from previous knowledge is an important component of grasping novel objects. On the other hand, Goodale et al. [9] showed that there is a dissociation between recognizing objects and grasping them, i.e., there are separate neural pathways that recognize objects and that direct spatial control to reach and grasp the object. Thus, given only a quick glance at almost any rigid object, most primates can quickly choose a grasp to pick it up, even without knowledge of the object type. Based on these two studies, a grasping algorithm should be able to grasp novel objects even without recognizing them, using previous knowledge. Our work represents a first step towards designing such an algorithm.

A handle is a part of an object made specifically to be grasped or held by the hand. Objects we use for everyday tasks are equipped with a part designed specifically to make

S. Khoury is with LISIF, Université Pierre et Marie Curie, Paris 6, France s.khoury@lisif.jussieu.fr

A. Sahbani is with ISIR, Université Pierre et Marie Curie, Paris 6, France anis.sahbani@upmc.fr

their grasp easier. Thus they have a handle. We all agree that the handle of a cup or a mug is its curved part. But what about the handle of a bottle, a pencil, a toothbrush or a spoon etc.? A further look to these objects shows that all of them have elongated parts to facilitate their grasp. Therefore, by identifying the handle of an object, one can obtain a successful grasp of it. We propose a learning approach that permits to predict an object handle as a function of the object's subparts.

Given a geometric model of an object, the proposed approach identifies its grasping part (Fig. 1). First, objects are decomposed into single parts. The learning step permits to perform an analogue of human choice of the grasping component. This part is called the handle of the object. At this point a human-like grasp of an unknown object can be obtained by computing n-finger force-closure grasps on its handle.

The idea of object handles was introduced in our previous work [24]. This paper aims to compute robust grasps of unknown objects in two steps: identifying their handles and generating stable grasps on them.

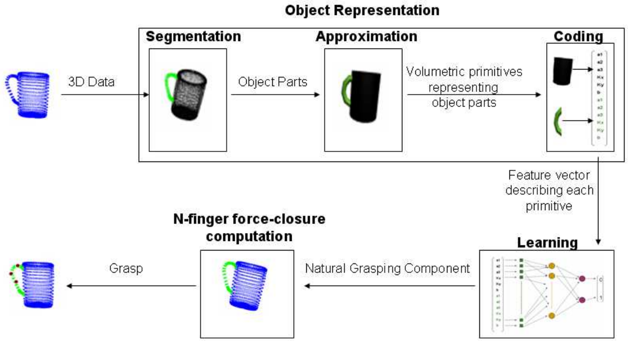
Object Representation
Segmentation
Approximation
Coding
3D Data
Object Parts
Volumetric primitives representing object parts
Feature vector describing each primitive
N-finger force-closure computation
Learning
Grasp
Natural Grasping Component

Fig. 1. The different steps of the proposed approach.

III. Object Representation

By representing objects as an assembly of geometric components, the proposed algorithm should be able to learn their grasping parts. Beginning with a complete 3D surface composed of triangle meshes, this paragraph details the different steps of the object representation. In the sequel, this study does not include the influence of the acquisition of the 3D points on the method.

A. Object coordinate system

We need a representation of the object that is invariant to its position and orientation. Input 3D points are expressed in the world coordinate system R_w. Therefore, before segmenting the object, these points are moved to the object centered coordinate system R_o with a homogeneous coordinate transformation T^{-1} [11].

$$\left(\begin{array}{cccc} x_o & y_o & z_o & 1 \end{array} \right)^T = T^{-1} \left(\begin{array}{cccc} x_w & y_w & z_w & 1 \end{array} \right)^T$$

Transformation T^{-1} is the inverse of transformation matrix T, which first rotates a point with a rotation matrix R and then translates it from the origin of the world coordinate system for $[t_x, t_y, t_z, 1]$. The vector $[t_x, t_y, t_z]$ represents the center of gravity t of all 3D points in the world coordinate system. To compute the rotation matrix, we compute first the matrix of central moments M:

$$\mathbf{M} = E[(\mathbf{X} - \mathbf{t})(\mathbf{X} - \mathbf{t})^T] \qquad (1)$$

where X is a 3D point of the object in the world coordinate system. The rotation matrix is the one that makes M diagonal. The columns of R are then eigenvectors of M.

B. Object Segmentation

Starting from a 3D surface model, a part decomposition step is performed to segment the object into its constituent single parts. A triangulation step is needed if only unstructured 3D point clouds are provided [10].

1) Segmentation based on Gaussain curvature and concaveness estimation: Segmentation is more common in the image processing area and has been recently introduced into the 3D mesh area. Many proposed segmentation algorithms use Gaussian curvature analysis to break down an existing structure into meaningful connected sub-components [22], [30]. These algorithms are efficient for certain 3D models. However, they cannot successfully process a high resolution 3D model as the geometric characteristics of the polygons in such a model, such as normals, curvatures are so close that these algorithms fail to detect some feature parts.

More recently, a 3D mesh watershed-based segmentation algorithm using Gaussian curvature and concaveness estimation have been proposed [5]. 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 a concaveness detection complement the Gaussian curvature. And more importantly, the algorithm enlarge the normal 1-ring neighborhood to an extended multi-ring neighborhood during the estimation of curvature and concaveness of each vertex in order to get more accurate geometric features for high resolution meshes. In other words, this algorithm is adapted for low resolution as well as for high resolution 3D models and thus we use it to decompose objects into meaningful parts. the next paragraph will give a brief description of this algorithm.

2) Watershed segmentation algorithm: After the Gaussian curvature and the concaveness estimation of each vertex on the 3D triangular mesh of the object model, vertices are labelled as boundary or inner region vertices. A vertex v is a boundary vertex if it has an hyperbolic behavior or is concave [5]. Watershed segmentation algorithms can be classified into two categories. One is the top-down flooding approach, which simulates that water floods from the ground to the local minima (also known as catchment basins). Another is the bottom-up immersion-based approach. Imagine that basins fill up with water from the local minima and at points where water coming from different basins would meet, watersheds are built [23]. The second approach is used and described 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 [5].

C. Approximation

The segmentation step of the algorithm decomposes an unknown object into its constituting single parts. The approximation step permits a geometrical description of these parts. Each part is represented by a superquadric, for their ability to describe a large variety of solids with only few parameters. Thus, the learning process will dispose of a compact geometric representation of the object components.

1) Superquadrics: Superquadrics are a family of geometric solids, which can be interpreted as a generalization of basic quadric surfaces and solids. They have been considered as volumetric primitives for shape representation in computer graphics [1] and computer vision [21]. Indeed, from one hand, they are convenient part-level models that can further be deformed and glued together to model articulated objects. From the other hand, with only a few parameters, superquadrics can represent a large variety of standard geometric solids as well as smooth shapes.

A superquadric surface model is defined by the following implicit equation:

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

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. 2). When both ϵ_1 and ϵ_2 are 1, the surface vector defines an ellipsoid or, if a_1, a_2, and a_3 are all equal a sphere. When $\epsilon_1 \ll 1$ and $\epsilon_2 = 1$, the superquadric surface is shaped like a cylinder. Boxes are produced when both ϵ_1 and ϵ_2 are $\ll 1$. Modelling capabilities of superquadrics can be enhanced by deforming them in different ways. In order to increase the flexibility of the model (2), we add two deformations : tapering and bending (Fig. 3).

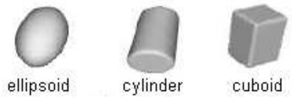

Fig. 2. Simple superquadrics.

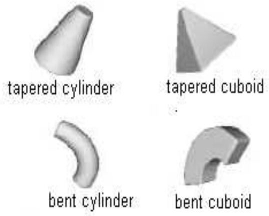

Fig. 3. Deformed superquadrics.

More details on the deformation parameters are provided in [28]. If we take into account these deformations, a superquadric can be modeled by 15 parameters; a_1, a_2, a_3 define the superquadric size; ϵ_1, ϵ_2 are for shape; k_x, k_y for tapering; k, α for bending; ϕ, θ, ψ for orientation; and p_x, p_y, p_z for position in space. We will refer to the set of all model parameter values as:

$$\lambda = \{a_1, a_2, a_3 \ldots a_{15}\} \quad (3)$$

In order to have a manageable number of superquadrics shapes, we have chosen 7 representative models (Fig. 2 and 3) that span the space of superellipsoids by choosing ϵ_1 and ϵ_2 to be one of 0.1 or 1.0.

2) Recovery of superquadric models: Given a set of N 3D surface points, we want to model them with a superquadric. We need to vary the 15 parameters a_j, j = 1, . . . , 15 in (3) to get such values for a_j that most of the 3-D points will lay on, or close to the model surface. Finding the model λ for which the distance from points to the model is minimal is a least-squares minimization problem. The minimization is performed with the Levenberg-Marquardt algorithm [15] which consists in a non-linear regression approach. The details of the approach used can be found in [28].

D. Object Coding

Many objects with similar components are grasped in the same manner. Bags, buckets, mugs and cups are composed of a cylinder and a curved cylinder. All these objects are grasped by their curved component, that is their handles. On the other hand, although a pocket-watch is composed of a cylinder and a curved cylinder, we do not grasp it by the curved part, it is relatively small to the non-curved one. Thus, we believe that the shape and the size of the object constituting parts are pertinent to the choice of the grasping component of an object. Therefore, we are interested in coding the shape and the size of these parts.

A superquadric is completely described by 15 parameters (3). But only 8 parameters (e_1, e_2, a_1, a_2, a_3, k_x, k_y and γ) are sufficient to represent the shape and the size of a superquadric. The other 7 parameters encodes the position and orientation of the superquadric. Therefore, a 8xS 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.

IV. Learning the grasping component

We have shown previously that the choice of the grasping component of an object is influenced by the shape and the size of the object constituting parts. Therefore, the proposed algorithm learns to use object components shapes and sizes in order to select the grasping part. Supervised learning [2] is used for this task, with synthetic objects (generated using computer graphics) as training data. Once the grasping part is determined, contact points are generated on the concerned part in order to ensure a force-closure grasp.

A. Training Data

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 perfectly synthetic data is easier and less-time consuming. Therefore, we use synthetic 3D models available on Princeton Benchmark [27] and NTU 3D Model Benchmark [6] along with labels indicating the grasping component. Since the learning algorithm should perform an analogue of human choice of the grasping component, different subjects were asked to identify the grasping part of the generated objects.

The training objects are the result of the assembly of two volumetric primitives. Our supervised learning requires a set of objects that can potentially span the space of two superquadrics assembly. Therefore, the choice of the training objects is important to effectively sub-sample this space. We use 12 objects for the training set (Fig. 4). We mentioned previously that 7 superquadrics will be used to model our objects. Thus, the training objects components are chosen to span these 7 superquadrics shapes with different sizes. Figure (5) 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 according to human 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. We use 36 of these examples for constituting our training data and the other 36 examples for constituting the testing data.

B. Learning Algorithm

A multi-layer perceptron, with one hidden layer, is trained with a typical backpropagation learning algorithm [2] in order to select the grasping part of a two-component object. We have shown previously that eight parameters are sufficient to represent a component. In the sequel, the first layer has sixteen inputs. 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 sufficient for obtaining a score of 100% for the training as well as for the testing data.

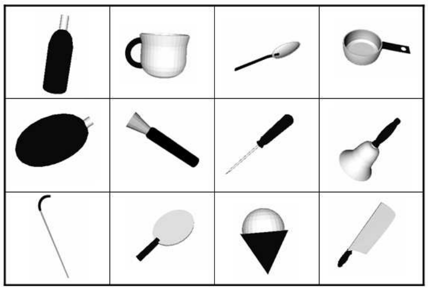

Fig. 4. Training objects set. The black part indicates the object handle.

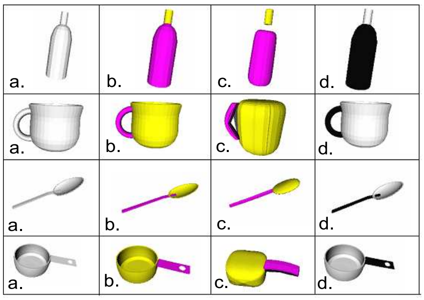

Fig. 5. Some two-part objects used for generating the training set. (a) shows the initial 3D object. (b) presents its segmentation into single parts. (c) shows the superquadric approximation of each constituting part. (d) shows the natural grasping part.

For multi-part objects, the decision of the grasping component is taken by considering the object parts two by two. 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.

C. Experimental Results

Different experiments were conducted to test the ability of the learning algorithm to generalize. First, we tested the algorithm on objects belonging to the same categories as the training data but of different shapes and sizes. These objects are such as bottles, spoons, forks, mugs, knifes, pencils etc. The motivation behind this experiment is that if our algorithm does not work on objects similar to the training data, then we must conclude that our feature set is not sufficiently discriminative. Fortunately, for such objects, the algorithm generalizes very well and was capable of finding each time the handle that human choose to grasp the corresponding object.

In a second time we tested the algorithm on multi-part objects that are completely different form those of the training set. This experiment is useful to test the ability of the algorithm to generalize to novel objects. Four subjects were asked to grasp 36 different objects in order to accomplish a task.

Thus the subjects were supposed to identify objects handles. Twenty objects, AO ($Agreed\ Objects$), were grasped by the same manner. In other words, the four subjects totally agreed on the objects handles. On the other hand, the remaining 16 objects, CO ($Confusing\ Objects$), induced confusion and the four subjects chose different parts to grasp them.

We tested then our algorithm on these objects. The system succeeds to find handles for 16 objects among the AO, which corresponds to a successful grasp rate of 80%. In other words, this rate shows that features such as sizes and shapes of an unknown object subparts are 80% discriminative to determine an object handle. Some of the AO objects are shown in (Fig. 6) along with their handles in black.

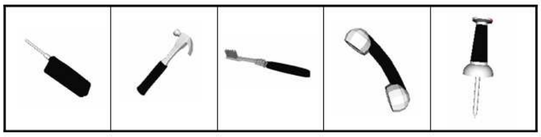

Fig. 6. Examples of AO objects. The black part indicates the corresponding object handle.

As for the CO objects, the grasping part selected by the system was always chosen by at least one person. Figure (7) 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 humans choice.

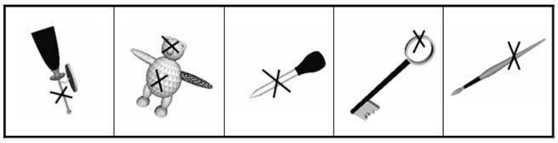

Fig. 7. Examples of CO objects. The black part indicates the system choice. The cross-marked parts indicate humans choice.

V. N-finger Force-Closure Grasp

The previous paragraphs detailed the selection of the NGC of an unknown object. This section concerns the generation of force-closure grasps on the selected part. For this purpose, we propose a new sufficient condition for N-finger force-closure grasps computation.

Force-closure property characterizes the stability of a grasp. According to the definition of Salisbury [26], a grasp is force-closure if and only if any external wrench can be balanced by the wrenches at the fingertips. This condition is equivalent to that the origin of the wrench space lies strictly inside the convex hull of the primitive contact wrenches [16], [17]. In the past few years, several force-closure tests were also proposed [14], [31]. These methods require considerable computation time. Researchers used heuristic approaches to improve performance by randomly generating grasps and filtering them [3] or by generating grasps in respect of specific rules which conducts to a necessary but not sufficient condition of force-closure [18].

We propose a sufficient but not necessary method to compute force-closure grasps of 3D objects. Our approach works with general objects and with any number n of contacts ($n \geq 4$). The main advantage of the method is its fast computation of force-closure grasps (5 times faster than the convex-hull [25]). This result is due to the formulation of the force-closure test as an inverse matrix multiplication. The method was detailed in [25]. The next paragraphs will give a brief review of the force-closure sufficient condition formulation and the advantage of using it for generating contact points on the NGC of an object.

A. A sufficient condition for N-finger force-closure grasps generation

As $3D$ force-closure grasps involve $6D$ wrench space. With a mere change of mathematical representation, using Grassmann algebra and Pl$\ddot{u}$cker coordinates, a $6D$ contact wrench can be represented by the line of action of its corresponding force. We use this mapping to prove that wrenches associated to three non-aligned contact points are of rank 6, thus form a basis of the $6D$ wrench space. This result induces the formulation of a sufficient condition for N-finger ($N > 3$) force-closure grasps.

Proposition 1: The 6 lines on the sides of a tetrahedron are independent, and thus form a basis of R^6, (Fig. 8).

Proof. To deal with lines in $3D$-space, we need a 4-dimensional linear space. For a basis of this space we can either take a point, O and 3 vectors e_1, e_2, e_3 or 4 points (p_0, p_1, p_2, p_3). We can relate these by:

$$p_1 = O;\ p_2 = O + e_1;\ p_3 = O + e_2;\ p_4 = O + e_3$$

Any point can be written as a linear combination of these 4 points, for example:

$$P_a = a_1 p_1 + a_2 p_2 + a_3 p_3 + a_4 p_4$$

where the a_i are scalars and the sums of the a_i are unity.

Lines are represented in Grassmannian terms by exterior products of points. Hence from these 4 independent basis points we can construct 6 independent lines which intersect to form a tetrahedron :

$$L_1 = p_1 \wedge p_2;\ L_2 = p_1 \wedge p_3;\ L_3 = p_1 \wedge p_4$$
$$L_4 = p_2 \wedge p_3;\ L_5 = p_2 \wedge p_4;\ L_6 = p_3 \wedge p_4$$

Any line L is now able to be represented as a linear combination of these 6 basis lines. We can explicitly display this by multiplying out and simplifying the exterior product of two points P_a and P_b on the chosen line:

$$L = P_a \wedge P_b = (a_1 p_1 + a_2 p_2 + a_3 p_3 + a_4 p_4)\wedge$$
$$(b_1 p_1 + b_2 p_2 + b_3 p_3 + b_4 p_4)\ \blacksquare$$

Proposition 2: Wrenches associated to 3 non-aligned contact points are of rank 6.

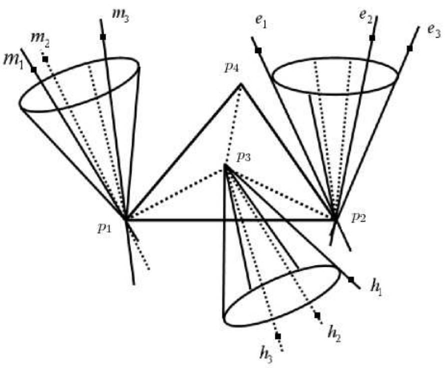

Fig. 8. The wrenches of rank 3 associated to the frictional contact points p_1, p_2 and p_3.

Proof. Let $p_1,\ p_2\ and\ p_3$ be 3 non-aligned contact points. Consider the friction cone associated to p_1, called CP_1 (Fig. 8). Let $\{m_1,\ m_2,\ m_3\}$ be three points chosen on any 3 non-coplanar lines of this cone. The lines $\{l_1 = p_1 \wedge m_1,\ l_2 = p_1 \wedge m_2,\ l_3 = p_1 \wedge m_3\}$ are of rank 3 [25]. Thus any line that passes through p_1 can be expressed as a linear combination of these 3 lines. Similarly, $\{e_1,\ e_2,\ e_3\}$ and $\{h_1,\ h_2,\ h_3\}$, are associated respectively to the friction cones $CP_2,\ CP_3$ at $p_2,\ p_3$. In the same manner, $\{l_4 = p_2 \wedge e_1,\ l_5 = p_2 \wedge e_2,\ l_6 = p_2 \wedge e_3\}$ and $\{l_7 = p_3 \wedge h_1,\ l_8 = p_3 \wedge h_2,\ l_9 = p_3 \wedge h_3\}$ are either of rank 3. Let p_4 be a point non-coplanar with $p_1,\ p_2,\ p_3$, so these 4 points constitute a tetrahedron.

The lines $(p_1 \wedge p_2)$, $(p_1 \wedge p_3)$ and $(p_1 \wedge p_4)$ can be expressed as a linear combination of $\{p_1 \wedge m_1,\ p_1 \wedge m_2,\ p_1 \wedge m_3\}$ since they all pass through p_1, thus:

$$p_1 \wedge p_2 = \sum_{i=1}^{3} \alpha_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \alpha_i l_i$$
$$p_1 \wedge p_3 = \sum_{i=1}^{3} \beta_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \beta_i l_i$$
$$p_1 \wedge p_4 = \sum_{i=1}^{3} \gamma_i (p_1 \wedge m_i) = \sum_{i=1}^{3} \gamma_i l_i$$

In the same manner, the lines $(p_2 \wedge p_3)$ and $(p_2 \wedge p_4)$ can be expressed as a linear combinations of $\{p_2 \wedge e_1,\ p_2 \wedge e_2,\ p_2 \wedge e_3\}$ since they pass through the contact point p_2. Finally the line $(p_3 \wedge p_4)$ passes through p_3 and thus can be expressed as a linear combination of $\{p_3 \wedge h_1,\ p_3 \wedge h_2,\ p_3 \wedge h_3\}$.

Since the lines of the tetrahedron are of rank 6 (from *proposition* 1*)*, they form a basis of R^6. We showed that the lines of the tetrahedron can be expressed as a linear combination of the 9 lines l_i. Thus these 9 lines, associated to the 3 friction cones, are also of rank 6. Consequently, a 6-dimensional basis can be extracted from these 9 lines. We remind the reader that the choice of 3 lines among the m sides of each linearized friction cone is due to the fact that these m lines are of rank 3.■

Proposition 3: 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 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:

- $\gamma\ is\ inside\ the\ linearized\ friction\ cone$ $of\ the\ kth\ finger$ (4)
- $\gamma = \sum_{i=1}^{k} \beta_i b_i,\ \beta_i < 0$

$$\Rightarrow\ \gamma = B\beta\ \ \Rightarrow\ \beta = B^{-1}\gamma \qquad (5)$$

where $B = [b_1, b_2, ..., b_k]$ is a $k \times k$ matrix and $\beta = [\beta_1, \beta_2, ..., \beta_k]^T$ 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 kth 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 [26]. A set of $k+1$ vectors in R^k positively span E^k if and only if the kth vector is a unique linear combination of the other k vectors and all coefficients are strictly negative [29]. The $k+1$ vectors $\{\gamma, b_1, b_2, .., b_k\}$ satisfy these conditions and thus positively span R^k. ■

B. N-finger force-closure grasps synthesis

To achieve force-closure, the grasp matrix should positively span the wrench space (*proposition* 1). Our method generates first, randomly, locations of $n-1$ fingers. We showed that wrenches associated to any three non-aligned contact points of 3D objects form a basis of their corresponding wrench space (*propositions* 2). Thus, we find all basis from the wrenches associated to these $n-1$ contacts. A position of the nth finger is located such that an associated contact wrench can be uniquely expressed as a strictly negative linear combination of one of the basis (*proposition* 3). This approach permits to compute all grasp points on the object for the nth finger to achieve force-closure grasp with the other $n-1$ fingers.

The main advantage of the grasp synthesis approach proposed is its fast computation. This is due, on one hand, to the fact that it reduces the force-closure test to an inverse matrix multiplication (*proposition* 3). On the other hand, testing different nth points for force-closure does not require any new computation of basis. In the contrary, the convex-hull algorithm constructs a different convex-hull at each change of points. A detailed comparison between our computation time and that of the convex-hull test can be found in [25]. Since this paper considers generating grasps on objects handles, such grasp synthesis algorithm is well adapted. One can generate randomly $n-1$ contact points on the graspable part and find the nth contact point among the handle points instead of searching the whole object. Figure (9) shows the contact points generated on the graspable part of different objects.

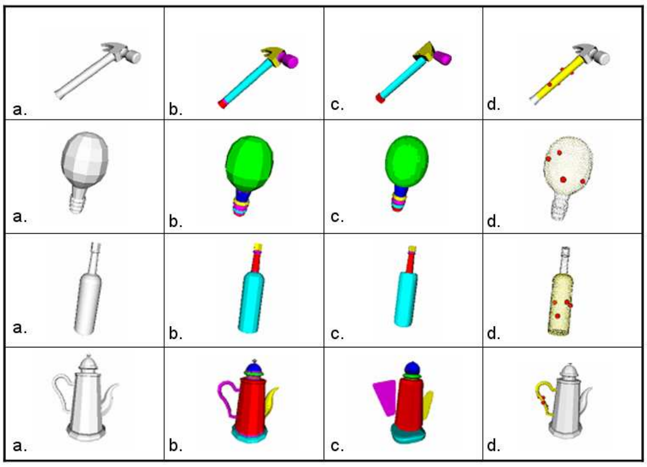

Fig. 9. (a)Shows the initial 3D object. (b) presents its segmentation into single parts. (c) shows the superquadric approximation of each constituting part. (d) Shows contact points on the object handle ensuring a force-closure grasp.

VI. Conclusion

Based on the idea that many objects are equipped with a handle to facilitate their grasp, we described an algorithm for computing robust grasps of unknown objects. Our approach imitates the human choice of the graspable part of the object. It predicts the object's handle as a function of features such as shapes and sizes of the object sub-parts. The proposed method is invariant to object translation, rotation and scale factor change. Experimental results conducted proved that the features selected are relevant to the choice of the graspable part of an unknown object and ensures a successful grasp rate of 80%. After the identification of the object handle, we presented a sufficient condition for generating n-finger force-closure grasps on it. A robust grasp of an unknown object is ensured on two-levels: grasp stability and task manipulation compatibility.

Our primary emphasis for the future will be to take into consideration the kinematic and geometric models of the robotic hand while generating force-closure grasps on objects handles.

References

[1] A. H. Barr, *Superquadrics and angle-perserving transformations*, IEEE Comput. Graphics Applicat., 1981, vol. 1, pp 11-23.

[2] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[3] Ch. Borst, M. Fischer and G. Hirzinger, *Grasping the dice by dicing the grasp*, IEEE Int. Conf. on Intelligent Robots and Systems, 2003.

[4] U. Castiello, *The neuroscience of grasping*, Nature Reviews Neuroscience, 6:726736, 2005.

[5] L. Chen and N. D. Georganas, *An efficient and robust algorithm for 3D mesh segmentation*, Multimedia Tools Appl. 29(2): 109-125, 2006.

[6] D. Y. Chen, X. P. Tian, Y. T. Shen and M. Ouhyoung, *On Visual Similarity Based 3D Model Retrieval*, Computer Graphics Forum (EUROGRAPHICS'03), Vol. 22, No. 3, pp. 223-232, Sept. 2003.

[7] S. Ekvall and D. Kragic, *Interactive grasp learning based on humain demonstration*, IEEE/RSJ International Conference on Robotics and Automation, New Orleans, USA, 2004.

[8] 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.

[9] M. A. Goodale, A. D. Milner, L. S. Jakobson, and D. P. Carey., *A neurological dissociation between perceiving objects and grasping them*, Nature, 349:154156, 1991.

[10] P.L. George and H. Borouchaki, *Triangulation de Delaunay et Maillage: Applications aux Elements Finis*, Hermes, Paris, 1997.

[11] B.K.P. Horn, *Robot Vision*, Cambridge, MA:M.I.T. Press, 1986.

[12] M. Hueser, T. Baier and J. Zhang, *Learning of demonstrated grasping skills by stereoscopic tracking of human hand configuration*, in Proc. IEEE Int. Conf. Robot. Automat., 2006.

[13] Y. Liu and N. S. Pollard, *A Sahpe matching algorithm for synthesizing humanlike envelopping grasps*, IEEE. Int. Conf. on Humanoid Robots, 2005.

[14] 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.

[15] K. Madsen, H. Nielsen and O. Tingleff, *Methods for non-linear least squares problems*, Technical University of Denmark, 2004.

[16] D. J. Montana, *The condition for contact grasp stability*, in Proc. IEEE Int. Conf. Robot. Automat., pp. 412-417, 1991.

[17] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical introduction to robotic manipulation*, Orlando, FL: CRC, 1994.

[18] N. Niparnan and A. Sudsang, *Positive span of force and torque components of four-fingered three-dimensional force-closure grasps*, IEEE Intl. Conf. on Robotics and Automation, 2007.

[19] E. Oztop and M. A. Arbib *Schema Design and Implementation of the Grasp-Related Mirror Neuron System*, Biological Cybernetics 87: (2) 116-140, 2002.

[20] R. Pelossof, A. Miller, P. Allen and T.Jebara, *An SVM learning appraoch to robotic grasping*, IEEE Intl. Conf. on Robotics and Automation, 2004.

[21] A.P. Pentland, *Perceptual organization and the representation of natural form*, Artif. Intell., 1986, vol. 28, no. 3, pp 293-331.

[22] S. Pulla, A. Razdan and G. Farin, *Improved curvature estimation for watershed segmentation of 3-dimensional meshes*, IEEE Trans. Visualization and Computer Graphics, 2001.

[23] JR. Roerdlink and A. Meijster, *The watershed transform: definitions, algorithms and parallelization strategies*, Fundamenta Informaticae 41: 187-228, 2001.

[24] S. Khoury, A. Sahbani and V. Perdereau, *Learning the natural grasping component of an unknown object*, IEEE Int. Conf. on Intelligent Robots and Systems, 2007.

[25] S. Khoury and A. Sahbani, *A Sufficient Condition For Computing N-Finger Force-Closure Grasps of 3D Objects*, to appear in IEEE International Conferences on Robotics Automation and Mechatronics, 2008.

[26] 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.

[27] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser, *The princeton shape benchmark*, Proc. of Shape Modelling International, Genova, Italy, 2004.

[28] F. Solina and R. Bajcsy, *Recovery of parametric models from range images: the case of superquadrics with global deformations*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, vol. 12(2), pp 131-147.

[29] R. Wagner, Y. Zhuang and K. Goldberg, *Fixturing faceted parts with seven modular struts*, IEEE International Symposium on Assembly and Task Planning, USA, 1995.

[30] Y. Zhang, A. Koschan and M. Abidi, *Superquadrics based 3D objects representation of automotive parts utilizing part decomposition*, Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, 2003, Vol. 5132, pp 241-251, Gatinburg.

[31] X. Zhu and J. Wang, *Synthesis of Force-Closure Grasps on 3D Objects Based on the Q Distance*, IEEE Transcations on robotics and Automation, vol. 19, no. 4, 2003.