Learning the Natural Grasping Component of an Unknown Object

Sahar El-Khoury, Anis Sahbani and Veronique Perdereau
Universite Pierre et Marie Curie - Paris 6,
LISIF, 3 rue Galilee, 94200 Ivry-sur-Seine, France
s.khoury@lisif.jussieu.fr

***Abstract*— A grasp is the beginning of any manipulation task. Therefore, an autonomous robot should be able to grasp objects it sees for the first time. It must hold objects appropriately in order to successfully perform the task. This paper considers the problem of grasping unknown objects in the same manner as humans. Based on the idea that the human brain represents objects as volumetric primitives in order to recognize them, the presented algorithm predicts grasp as a function of the object's parts assembly. Beginning with a complete 3D model of the object, a segmentation step decomposes it into single parts. Each single part is fitted with a simple geometric model. A learning step is finally needed in order to find the object component that humans choose to grasp it.**

I. INTRODUCTION

In the recent past, robots had only one role, working in factories to assume repetitive tasks. Nowadays, robots are finding their way into human living space. Many researchers are interested in developing humanoid robots to help people in their daily life. Such robots should be autonomous and able to interact with objects around them.

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. Carefully hand-programmed robots can execute amazing manipulation tasks, from using tools to assemble complex machinery to serving tea and other useful tasks [20]. However, fully autonomous grasping of a previously unknown object remains a challenging problem. 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.

II. RELATED WORK

In literature, different approaches have addressed the problem of grasping. Some methods considered grasping unknown objects problem. They proposed solutions based only on the geometric model of objects. The latter can be obtained by a stereovision or a 3D laser sensors. Others imitated human grasping by learning the contact points between the fingers and the object surface.

In [7], the authors generate a number of feasible grasp candidates by randomly choosing different contact points on the object surface. These candidates are evaluated according to a grasp stability measure. The best candidate is selected. In a similar approach, the authors, in [5], generate a set of starting grasp locations based, this time, on a simplified object description. Objects are modelled by a set of shape primitives, such as spheres, cylinders, cones and boxes. Feasibility and quality of these grasps are then determined. The best grasps are presented to the users. These approaches find stable grasps adapted to pick and place operations. However, they fail to determine suitable grasps for object manipulation.

Instead of generating and testing a set of grasps, different methods tried to compute optimal grasps. In [3] the authors propose a non-convex object grasp planner. The algorithm makes an iterative partition of the object into subcomponents using the Approximate Convex Decomposition process [4]. At each iteration, a set of grasps is computed for each generated component. Every grasp is ranked depending on a quality criterion. The grasp with the best quality is chosen as the output of the planner. In other words, the approach purpose is to find, from the object geometry, a component with a good grasp quality. Evidently, the chosen grasp may not be the appropriate one to the required task. An approach that tries to imitate human grasping is proposed in [2]. Based on the observation that humans preshape their hands to grasp an object, the authors present a solution to guide the grasp planning by finding a grasping axis from the object geometry called natural grasping axis (NGA). This axis is parallel to the hand palm during the preliminary approach (before grasping) and conditions the fingers positioning on the object surface. Michel et al., [2], considered that the grasping axis is one of a panel of the most parallel lines with all pair of facing facets of the object. The major limitation of the method is its inability to select the NGA from among the others.

Learning algorithms have also been applied to grasping problems. The paper [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 was proposed in [9]. 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. These approaches

enable objects telemanipulation but are not adapted to grasping unknown objects.
In the work presented here, we utilize past various results in the field of object representation using primitives to propose a novel strategy for the problem of grasp learning (Fig. 1). This strategy permits to grasp unknown objects in the same manner as humans.

III. Our Approach

In a fraction of a second, humans are able to grasp novel objects. To account for this extraordinary capacity, we take inspiration from the Recognition By Components (RBC) theory [1]. RBC explains how novel objects are successfully recognized by the visual system. It proposes that the visual system extracts geons (or geometric ions such as cubes, spheres, cylinders) and uses them to identify objects. It also proposes that representations of objects are stored in the brain as structural descriptions that contain a specification of the object geons and their interrelations. If people represent objects as an assembly of geometric primitives to recognize them, why not suppose that they use the same strategy to grasp them?
We believe that many objects are equipped with a part designed specifically to make their grasp easier, when performing everyday tasks. For example, mugs, cups, suitcases, domestic irons have curved parts, that is their handles. Pencils, bottles, toothbrushes, spoons have elongated parts. Given a geometric model of an object, the proposed approach identifies its grasping component. Thus, objects are decomposed into single parts. A learning step permits to perform an analogue of human choice of the grasping component. The algorithm is therefore called Learning the Natural Grasping Component. Since learning algorithms expect a fixed length feature vector, a geometrical representation of the object parts is required. For that reason, an approximation and object coding steps are needed. Once the grasping part is found, the Natural Grasping Axis [2] of the object is easily extracted by computing the inertial axes of the concerned component. We remind that this axis is parallel to the hand palm during the preliminary approach (before grasping) and conditions the fingers positioning on the object surface. The diagram below (Fig. 1) shows the different steps to find the Natural Grasping Component (NGC) of an unknown object. The following paragraphs will detail each step of the algorithm.

IV. Object Modelling

By representing objects as an assembly of geometric components, the proposed algorithm should be able to learn their Natural 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.

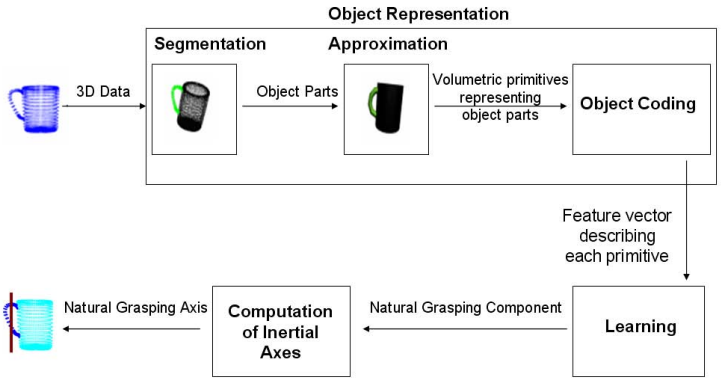

Fig. 1. The different steps of the approach.

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} [13].

$$\begin{pmatrix} x_o & y_o & z_o & 1 \end{pmatrix}^T = T^{-1} \begin{pmatrix} x_w & y_w & z_w & 1 \end{pmatrix}^T$$

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:

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

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 [21].
The part decomposition algorithm used [12] is based on curvature analysis and consists of three major steps, Gaussian curvature estimation, boundary detection and region growing.

1) Gaussian curvature estimation and boundary detection: Beginning with a complete 3D object model composed of triangle meshes, a Gaussian curvature is estimated for each vertex of a triangle mesh [14]. Gaussian curvature of the vertex p is computed as :

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

where :

- N, 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.

A specified threshold is then applied to label vertices as boundary or seed. Vertices of highly negative curvature are labelled as boundaries between two parts while the rest are labelled as seeds belonging to potential object parts. This threshold is determined in a heuristic way depending on object and mesh resolution.

2) Region growing: After the vertices are labelled, a region-growing operation is performed on each vertex labelled as seed. The process terminates when the grown regions are surrounded by boundary vertices. Fig. 2 shows the decomposed parts of the teapot. Each region is represented with a different color.

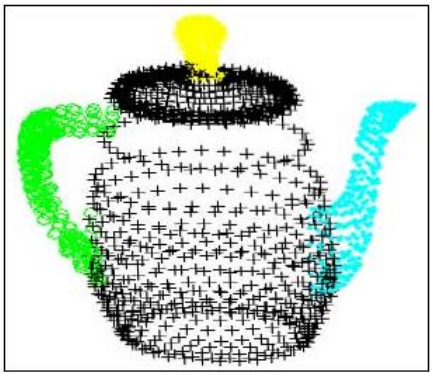

Fig. 2. Segmentation of a teapot.

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 [10] and computer vision [11]. 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:

$$f(x,y,z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1 \quad (3)$$

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.

Based on this implicit equation of the superquadric surface we define the following function:

$$F(x,y,z) = f^{\epsilon_1} \quad (4)$$

This compact model of superquadrics, defined by only five parameters, can model a large set of building blocks like spheres, cylinders and boxes (Fig. 3). 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

Fig. 3. Simple superquadrics.

capabilities of superquadrics can be enhanced by deforming them in different ways. In order to increase the flexibility of the model (3), we add two deformations : tapering and bending (Fig. 4).

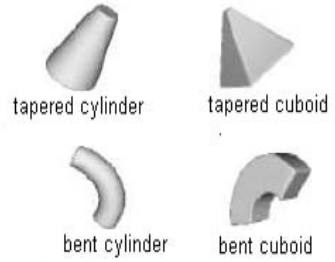

Fig. 4. Deformed superquadrics.

- Tapering is defined by two parameters k_x and k_y.
- Bending is defined with the two parameters k and α. k is the curvature parameter and α determines the bending plane. Knowing these two parameters, the bending angle, γ, can be easily computed.

More details on the deformation parameters are provided in [15]. 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:

$$\lambda = \{a_1, a_2, a_3....a_{15}\} \tag{5}$$

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 (5) 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 [15]. For each point, the following distance is calculated :

$$d = F - 1 \tag{6}$$

This distance is then minimized for the N points:

$$min \sum_{k=1}^{N} d_k^2 \tag{7}$$

The minimization of is performed with the Levenberg-Marquardt algorithm [16] which consists in a non-linear regression approach. Initial estimate of the set of model parameters λ determines to which local minimum the minimization procedure will converge. Only very rough estimates of object true position, orientation, and size suffice to assure convergence to a local minimum that corresponds to the actual shape. Initial values for both shape parameters, ϵ_1 and ϵ_2 can always be 1, which means that the initial model is always an ellipsoid. To compute the position and the orientation of the object centered coordinate system, we compute the matrix of central moments (1). k_x, k_y and α are initialized to 0 and k to a very small value which corresponds to a nondeformed model.

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 (5). But only 6 parameters (a_1, a_2, a_3, k_x, k_y and γ) are sufficient to represent the shape and the size of a superquadric. ϵ_1 and ϵ_2 are not taken into account as they only determine the shape roundness (form square to cylinder). Thus, a 6xS column vector V, where S is the object part number, represents the whole object. The S superquadrics are labelled from 1 to S with respect to their volume. The superquadric labelled 1 is the most voluminous one and thus the first one to be represented in the vector V. 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.

V. 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 is used for this task [19], with synthetic objects (generated using computer graphics) as training data. Once the grasping part is determined, the Natural Grasping Axis of the object is considered as the principal axis of the concerned part.

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 generate synthetic objects along with labels indicating the grasping component (Fig. 7). 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 volumetric primitives used to create objects are from different sizes and shapes. As mentioned previously, shape roundness is not considered in the object coding, thus ϵ_1 and ϵ_2 can be set to any constant values we choose. For example, when $\epsilon_1 = 0.1$ and $\epsilon_2 = 1$, the superquadric surface is shaped like a cylinder. Fig. 5 shows the non-deformed primitives chosen. To enlarge the flexibility of object representation, two deformations, tapering and bending, are added to each primitive (Fig. 6). The training objects are the result of the assembly of two volumetric primitives (Fig. 7).We generate 56 objects examples. 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. The time consuming part of synthetic data generation is the manual creation of the numerical models of the object. A Perceptron Learning Algorithm is then trained in order to predict from a two-component object the grasping part [19].

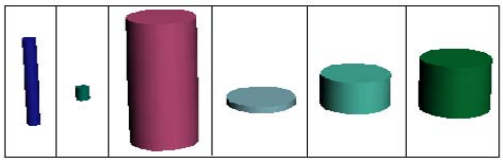

Fig. 5. Non-deformed primitives chosen to create objects for the training set

B. Grasping Multi-part Objects

A multi-layer perceptron, with one hidden layer, is trained with a typical backpropagation learning algorithm in order

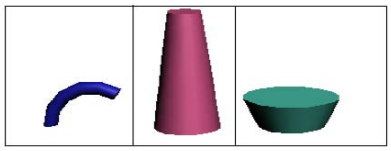

Fig. 6. Examples of deformed primitives chosen to create objects for the training set

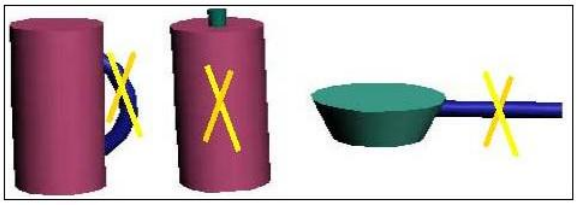

Fig. 7. Examples of the training set objects, the grasping component is marked with a cross

to select the grasping part of a two-component object. We have shown previously that six parameters are sufficient to represent a component. In the sequel, the first layer has twelve 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. For multi-part objects, the decision of the grasping component is taken by considering the object parts two by two starting with the most voluminous ones. In other words, the algorithm starts by choosing a grasping component between the most two voluminous parts of the object. The chosen part is then compared with another component and so on until finding the NGC of the multi-part object.

C. Prehension Axes

Once the grasping component is determined, its principal axes are computed. Therefore, the matrix of central moments, M, is calculated (1). M has three eigenvectors. Eigenvector e_l with the smallest eigenvalue λ_1 corresponds to the minimum inertia line and the eigenvector e_2 with the largest eigenvalue to the maximum inertia line. The minimum-inertia line is known as the principal axis and thus considered as the NGA of the geometric part in question [2].

VI. Results

We first tested the algorithm for its capability of finding the grasping part of synthetic objects not in the training set. The accuracy was 99%. Next, we tested the algorithm on many objects (Fig. 8) for which 3D models are available on Princeton Benchmark [17] and NTU 3D Model Benchmark [18]. We performed experiments on 2 part objects such as cups, pencils, spoons, bottles, toothbrushes, all of different sizes and appearances of the ones in the training set, as well as a large set of multi-part objects such as wineglasses, teapots, pens, cell-phones, pocket-watches, etc.

Figures 9 and 10 show the different steps of the algorithm tested on a mug and a teapot. The mug is first segmented into two parts (Fig. 9). These parts are then approximated by the convenient superquadrics: a cylinder and a curved cylinder. Therefore, the vector V_m that encodes the mug is a 6x2 column vector. The first 6 parameters of V_m encode the cylinder and the other 6 parameters describe the second part, the curved cylinder. The same procedure is used to represent the teapot (Fig. 10). This time, a 6x4 column vector V_t encodes the 4 parts of the teapot. The graspable component of the two objects found by the algorithm, are marked with a cross.

In extensive experiments, the algorithm appeared to generalize very well. It was usually able to identify the correct grasping component of real objects (rather than synthetic), including many different from ones in the training set. The algorithm was also tested on different examples of various sizes and shapes of the same object type. Tables I & II show the grasp rate obtained for each object type. The grasp rate indicates the amount of success in finding the grasping component. We note that the grasping component of an object is the one that humans employ to grasp the object.

TABLE I

Grasp Rate for two-part real objects

Tested on	Grasp Rate
Mugs	80%
Bottles	100%
Buckets	80%
pencils	100%
spoons & forks	70%
toothbrushes	100%

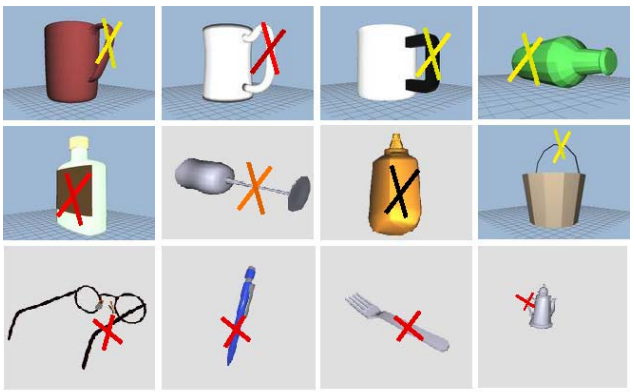

Fig. 8. Some real objects on which the algorithm was tested, the grasping component is marked with a cross

For objects such as pencils, pens, toothbrushes, cell-phones, wine-glasses etc..., the algorithm performed perfectly, (grasp rate of 100%). However, for objects such as mugs, teapots, buckets, the algorithm has a lower success rate. This low rate is not mainly attributed to the learning algorithm but to the approximation step. The latter failed to approximate some of the handles of these objects with a

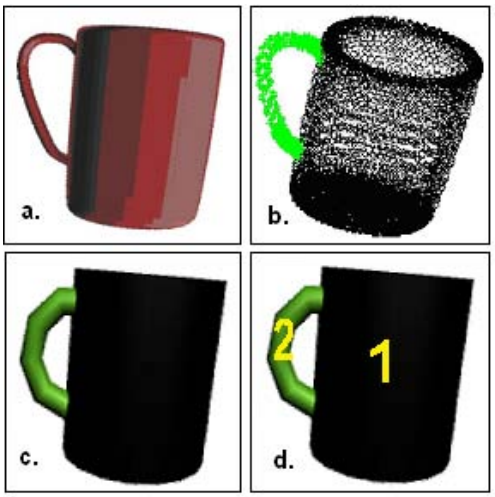

Fig. 9. a) 3D model of a mug. b) The mug segmentation into two parts. c) The superquadric representation of each part. d) Superquarics labelling in respect to their volume.

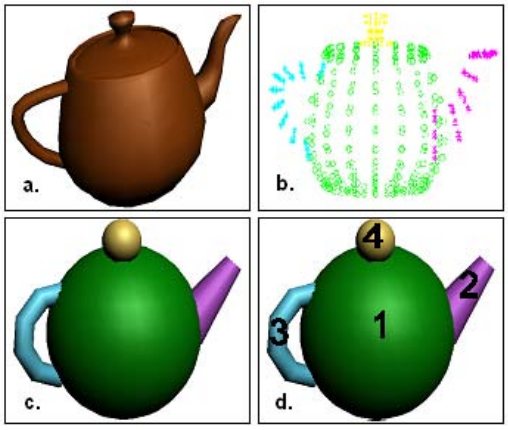

Fig. 10. a) 3D model of a teapot. b) The teapot segmentation into four parts. c) The superquadric representation of each part. d) Superquarics labelling in respect to their volume.

curved model. In the sequel, they were not chosen by the learning algorithm as the grasping component. However, the algorithm was able, in many instances, to pick up completely novel objects by identifying their grasping part.

VII. CONCLUSIONS

Based on the idea that many objects are equipped with a part designed specifically to make their grasp easier, we described an algorithm for identifying the Natural Grasping Component (NGC) on a previously unknown object. Our algorithm predicts the NGC of an object as a function of the assembly of the object parts. The approach is invariant to object translation, rotation and scale factor change. Despite being tested on objects of different shapes and sizes than the ones in the training set, the algorithm generalizes very well. It succeeds in finding the NGC of many novel objects which proves the effectiveness of the proposed idea.

TABLE II

GRASP RATE FOR MULTI-PART REAL OBJECTS

Tested on	Grasp Rate
Teapots	50%
Cell-phones	100%
wineglasses	100%
eye-glasses	100%
pens	100%

REFERENCES

[1] I. Biederman, *Recognition-by-Components: A Theory of Human Image Understanding*, Psychological Review, 1987, vol. 94, pp 115-147.

[2] C. Michel, C. Rimond, V. Perdereau and M. Drouin, *A robotic grasping planner based on the natural grasping axis*, Proceedings of the International Conference on Intelligent Manipulation and Grasping, Genoa, Italy, 2004.

[3] E. Lopez-Damian, D. Sidobre and R. Alami, *Grasp planning for non-convex objects*, 36th International Symposium on Robotics, ISR, Tokyo, Japan, 2005.

[4] J.M Lien and N. Amato, *Approximate convex decomposition*, technical report, TR03-001, PARASOL LAB, Texas A&M University, 2003.

[5] A.T. Miller, S. Knoop, H.I. Chritensen and P.K. Allen, *Automatic grasp planning using shape primitives*, Proc. IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003.

[6] N.S. Pollard, *Closure and quality equivalence for efficient synthesis of grasps from examples*, International Journal of Robotics Research, 2004, vol. 23 , pp 595-614.

[7] Ch. Borst, M. Fisher and G. Hirzinger, *A fast and robust grasp planner for arbitrary 3D objects*, Proc. IEEE International Conference on Robotics and Automation, Detroit, Michigan, 1999.

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

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

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

[12] 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.

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

[14] C. Lin and M. Perry, *Shape description using surface triangulation*, conf. Computer Vision: Representation and Control, 1982, pp 38-43.

[15] 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.

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

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

[18] 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.

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

[20] T. Shin-ichi, *Living with robots*, Nipponia, 2006.

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