A Decision Tree Based Approach to Grasp Synthesis

C. Fernandez, A. Vicente, O. Reinoso, R. Aracil*
Systems Engineering and Automation Division
Miguel Hernandez University
Av. Ferrocarril s/n
03202 Elche (Alicante) Spain

*UPM-DISAM
Polytechnical University of Madrid
C/ Jose Gutierrez Abascal, 2
28006 Madrid, Spain

E-mail c.fernandez@umh.es

Abstract

A generic learning based approach to robot grasp synthesis is presented. This generic approach can cope with different robot hands and different kinds of sensing information, as a difference with previous learning based approaches focused mainly on two jaw grippers and 2D information of the object to be grasped.

The proposed methodology structures the grasp synthesis in four steps. First, the outer points of the object (2D or 3D) are filtered using local information in order to select those valid as contact points, by means of a decision tree inferred from the examples. In a second step, all the sets of valid reachable points are computed taking into consideration the robot hand used. The third step selects the best set among all the previously computed so giving the optimum contact points for the grasp: a decision tree and a further nearest neighbour test working with global information are used for this purpose. In the last step, a search is performed in order to find the best hand configuration among those capable of reaching the optimum contact points.

Some experimental results are presented, showing how the system performs well after few training examples.

1 Introduction

Unlike traditional robotic applications, where robots perform repeatedly the same tasks under exactly the same conditions, new robotic applications place robots in unstructured changing environments and performing several different operations. In these situations, the robot must decide by itself how to grasp each object or tool depending on its geometry, its mass distribution, the operation to be performed with it, the environment, the gripper configuration, etc.

At present, robotic applications working in non structured environments and performing different operations with a variety of tools are usually performed through teleoperation. The tasks involved are usually very complex and are performed in dangerous or hostile environments. Some examples can be found in nuclear industry, underwater tasks, live power line maintenance, etc. [1][2]. Even though the robot is controlled by an human operator, partial automation of tasks in teleoperated environments [3] or collaborative control [4] (where the robots do not act as mere slave devices) are challenging research fields, and one of the tasks suitable for being automated is object grasping.

Apart from teleoperation, there is another field of application of robotics where robots have to work in unstructured environments and handling different parts and tools: service robotics. The International Federation of Robotics [5] defines a service robot as ‘a robot which operates semi or fully autonomously to perform services useful to the well being of humans and equipment, excluding manufacturing operations’. Some examples can be found in the Care-O-Bot [6] whose task is to help elderly or mobility impaired persons in their daily life; or the Manus-arm [7] which is side-mounted on a wheelchair and is able to open doors, handle food or drinks, etc. These robots are not just research prototypes: Manus-arm is a commercially available product with more than 100 units already working. Other field of application of service robots is entertainment where the robot QRIO [8] is the latest example of robot designed to interact with humans. A complete list of service robots can be found in [9].

All the mentioned examples of teleoperation and service robots share a common problem: the need to grasp different objects in unstructured environments. In this paper, a learning based approach to grasp automation is presented. The goal is to find a learning based approach generic enough in two aspects: first, in the kind of robotic hand or gripper used; and second, in the kind of sensorial information available. So, the same structure should be able to synthesize grasps in the simplest situation: a two jaw gripper and 2D contour information of the object to be grasped; and in very complex situations: multifingered robot hands and 3D information of the object to be grasped obtained from multiple range sensors.

As a difference with non learning based methods, a learning based system is able to imitate the user behaviour [17]. In this sense such grasp automation systems can learn to grasp an object by its handle from the examples given by the user even if such grasp is not optimal in terms of stability, force closure or any other criteria.

2 Previous approaches

Previous approaches to robot grasp synthesis can be grouped in two categories:

- Those not being learning based. Among these, there are several alternatives: synthesizing grasps meeting the force closure condition [10][11], synthesizing optimal grasps according to different quality criteria [12][13], synthesizing grasp starting from generalized prototypes [14], etc. There are two main drawbacks common to all these approaches: first, they rely on a precise knowledge of the 2D or 3D geometry of the object to be grasped, which is not available in a real grasping scenario; and second, they cannot cope with extra information apart from the geometry of the object (i.e. they can not consider the operation to be performed with the object) Apart from that, most methods are not feasible in a real time application as they are computationally intensive.
- Those being learned based. There are few such approaches, among them the one proposed by Kamon [15] or the one proposed by Schwammkrug [16]. These approaches are designed specifically for just one grasping situation –usually simple- where a particular two or three jaw gripper grasps an object taking into consideration only 2D information of its contour. In this sense, they are not general enough.

The review of previous approaches leads us to establish a set of requirements that a grasp synthesis method should meet in order to be applicable in a real scenario:

- It should not be computationally expensive.
- It should not require a complete knowledge of the 2D or 3D geometry of the objects.
- It should consider the operation to be performed with the object.
- It should be applicable to different robot hands and different grasping scenarios (2D, 3D).

3 Proposed approach

A learning based generic approach is proposed, where the training examples can be obtained by teleoperation or through a simulation environment (the goal is to learn from human demonstration).

Let us consider a generic n finger grasping hand with an undefined m_n number of joints per finger. A certain grasp of an object can be represented by the n contact points in the object plus the configuration of each finger (in general different configurations can reach the same contact points, the grasps not being equivalent). If the robot hand is attached to the wrist of an m DOF robot arm, then these m DOF are also relevant for the grasp. Equation 1 represents the set of n contact points, where S is the outer surface or contour of the object to be grasped, $k=2$ on a 2D scenario and $k=3$ on a 3D one; and equation 2 represents the whole set of configuration parameters both for arm and hand.

$$\{p_1, p_2, ..., p_n\} \quad p_i \in S \subset \Re^k \tag{1}$$

$$\{q_1, ..., q_m, q_{11}, ..., q_{1m_1}, ..., q_{n1}, ..., q_{nm_n}\} \tag{2}$$

This generic robot arm and hand is depicted in figure 1.

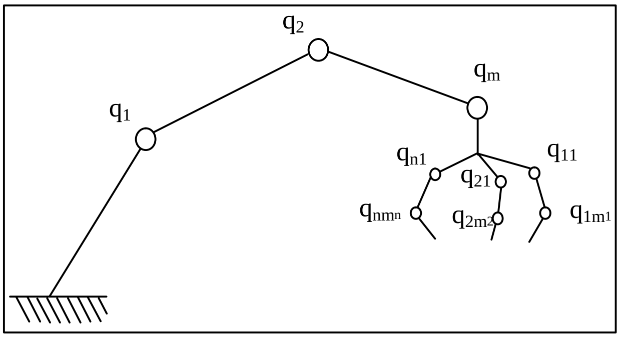

Figure 1: configuration parameters for arm and hand

In order to be able to cope with all kinds of grasping scenarios, the grasp synthesis is performed in four steps, which are described in the following paragraphs.

1. The first step consists on a filtering of the possible contact points in order to select only those appropriate for placing a finger on them. The set of all possible contact points includes all the contour points (2D) or surface points (3D) of the object. The filtering criteria are obtained from grasps examples provided by the user and by means of machine learning techniques. In the current implementation of the approach, a decision tree is generated from the examples to infer the filtering rules; and the data supplied to the decision tree algorithm includes the distance from each point to the center of mass of the object and a multirresolution measure of local convexity, as described in our previous work [17]. As a result of this first step, a subset of z' valid points is extracted from the original z contour points, where *z’*£ *z*. Only local data is processed in this step.

2. The second step computes all the sets of n grasping points reachable by the robot hand under consideration. This step is not learning-based and it is very dependant on each particular robot hand. Complexity increases with the number of fingers n; *num* in equation 3 represents the number of different sets to check.

$$num = \binom{z'}{n} \tag{3}$$

3. The third step is again learning based and its purpose is to select, among all the sets of reachable points, the best one according to the examples provided by the user. In the present implementation of the approach, two machine learning techniques are used: first, a decision tree is inferred from the examples to obtain validity rules and reduce the number of candidate sets; and then all the remaining candidate sets are compared with the stored examples using nearest neighbour techniques in order to select the most similar to one of the example grasps. Both algorithms work on the same data (global data): a distance measure from the center of mass of the object to the center of the convex space of all contact points (which will be used as a reference point) and multirresolution measurements of the angle between the normal at the contact point and the line directed to the reference point. The experiments carried out have shown that these data are enough to characterize a certain contact point set. The output of this step is the best set of *n* contact points.

4. The last step, once the contact points have been selected, deals with the selection of the best arm and hand configuration among those capable of reaching the *n* contact points. This step is again very dependant on the particular robot hand and is performed using a weighted version of the nearest neighbour algorithm, where higher weights are given to those joints closer to the contact point. In some simple situations, as those where the robot hand is a simple two jaw gripper, there is only one configuration capable of reaching a certain set of contact points, so step 4 can be avoided.

4 Computational complexity

Searching for the best set of contact points among all the outer points of an object and considering both local (related to a certain point of contact) and global (related to a set of contact points) data can be computationally very expensive. The four step approach proposed highly reduces the computational load.

A direct brute force approach will check both local and global data for all combinations of *n* contact points from the whole set of *z* surface points. Considering that local data is not recomputed several times for the same contact point, equation 4 reflects the total computational cost, where c_L represents the cost of local data computation and c_G represents the cost of global data computation.

$$c_{brute\_force} = z \cdot c_L + \binom{z}{n} \cdot c_G \qquad (4)$$

The proposed approach computes local data for all the *z* points but global data for only *z'* points (all but the ones filtered in the first step); the total cost is reflected in equation (5)

$$c_{proposed} = z \cdot c_L + \binom{z'}{n} \cdot c_G \qquad (5)$$

Both approaches perform only *z* local data computations, but the real problem is the number of global data computations which grows rapidly with the number of contact points. The complexity reduction in global computations offered by the proposed system can be expressed as a ratio *r*, as equation 6 shows. In this equation, the number *n* of robot fingers (which usually can vary from 2 to 5) has been considered neglectible with respect to the number of surface points before or after the filtering (*z* or *z'*).

$$r = \frac{\binom{z}{n}}{\binom{z'}{n}} = \frac{z!\,(z'-n)!}{z'!\,(z-n)!} \approx \left(\frac{z}{z'}\right)^n \qquad (6)$$

The ratio shows a complexity reduction ratio which is dependant on the degree of filtering of the first step; and whichs grows rapidly with the number of fingers. That is exactly the desired behaviour, as the computational complexity only becomes a problem when the number of fingers grows. Experimental results show that the degree of filtering of the first step reduces the number of surface points from 10 to 100 times, thus resulting in a complexity reduction that ranges from 10^n to 100^n.

5 Knowledge representation

Steps 1 and 3 of the proposed approach are learning based, that means that some examples are provided by the user and the system infers behaviour rules from them. In order to choose a knowledge representation method for such rules, both a quantitative and a qualitative analysis have been performed.

Concerning the quantitave analysis, steps 1 and 3 can be considered as classification problems: step 1 classifies surface points according to convexity and distance measurements; and step 3 –in a first instance- classifies sets of contact points according to similar measures. In order to perform a comparison, a database of 400 example grasps was generated. Among them, 200 were correct grasps performed by the user and the other 200 were wrong grasps generated randomly (random generation of bad examples is fully detailed in our previous work [17]). Two different tests were performed to check the classification accuracy of each method. The first test was a 10 fold cross validation using all the 400 examples (90% were training examples and the remaining 10% were test examples). 40 different tests were performed with each classifier in order to obtain statistical measures of classification accuracy. The classifiers used for the comparison were: multilayer perceptron (MLP), nearest neighbour (NN), decision trees (DT), and the naive Bayes classifier (NB). Each classifier was tuned to obtain the maximum performance over the examples, and the results in terms of percentage of correct classifications and standard deviation are shown on table 1. The results are also represented in figure 2 assuming a normal distribution for the data and adjusting its mean and standard deviation to those obtained in the tests.

Classifier	Correct classif. (%)	Stand. Deviation
NN	94.57	0.86
DT	91.25	0.96
MLP	88.7	1.38
RBF	83.19	2.24
NB	79.0	1.73

Table 1: cross validation results

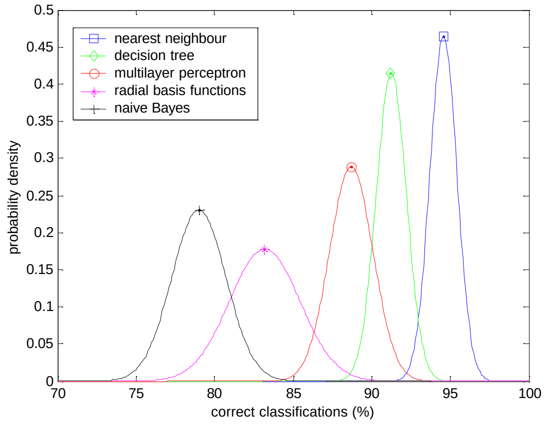

Figure 2: normal distribution of results

The results show clearly that nearest neighbour techniques give the better classification accuracy, followed by decision trees and MLP.

The second test compares the performance of the different classifiers according to the number of training examples used. Results were obtained for a number of training examples ranging from 20 to 380 and are shown (averaged after 40 repetitions) on figure 3.

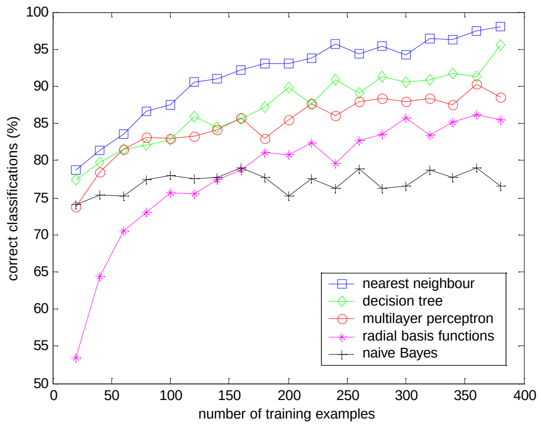

Figure 3: performance vs. training examples

This second test gives similar results: the best classification rates are obtained with nearest neighbour method followed by decision trees and MLP.

Concerning the qualitative analysis, decision trees are considered the best option as the model generated is readable and thus can be checked by the user if necessary (nearest neighbour techniques do not infer a model at all and MLP generates an unreadable model). Apart from that, decision trees are very robust to noise in the training data [18][19], and results are low dependant on parameter setting (the well known C4.5 algorithm [20] was used to generate the decision trees) as a difference with MLP where the number of hidden units is a critical parameter.

As a result of both quantitative and qualitative analysis, the option chosen was to use decision trees for step 1 and the first part of step 3 (selection of valid sets of contact points) and nearest neighbour techniques for the second part of step 3 (the valid set closer to the examples is chosen as the optimum set of contact points).

6 Experimental results

A simulation environment has been developed in order to check the behaviour of the proposed system. In this environment the user controls a robot arm and hand with a joystick in order to give grasping examples to the system. For this purpose, different objects are presented in the working area of the robot and the user must guide the robot to grasp each object. All the training examples are stored and, after a selectable number of examples, the system computes the decision trees and is able to perform grasp autonomously.

At present, the simulation environment is restricted to a SCARA robot (2D grasps) and a two jaw gripper. Figure 4 shows the full robot workspace and an object to be grasped, while figure 5 zooms into the details of the grasping of an object.

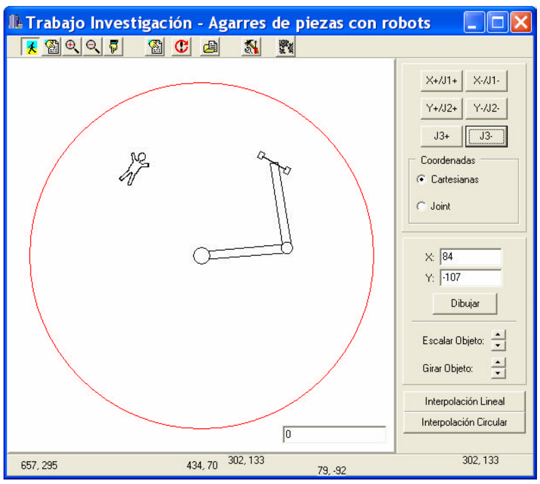

Figure 4: simulation setup

In order to perform the tests, a database of 24 different 2D objects has also been created. The complete set of objects is shown on figure 6. Several objects are mere geometrical shapes and should be grasped according to stability, force closure or similar criteria. Other objects, like the screwdriver, the pan, the corkscrew or the bottle are examples of grasps related to the operation to be performed with the object and therefore those grasps cannot be computed according to the previous criteria; the goal is to infer rules from the examples given by the user as to be able to grasp these particular objects correctly.

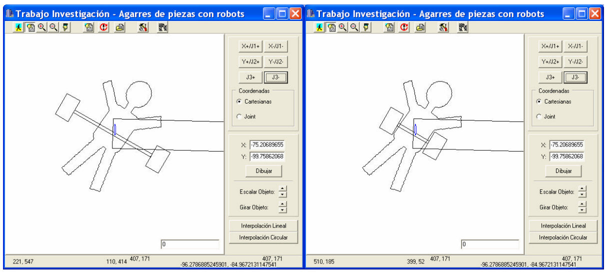

Figure 5: object grasping with the simulator

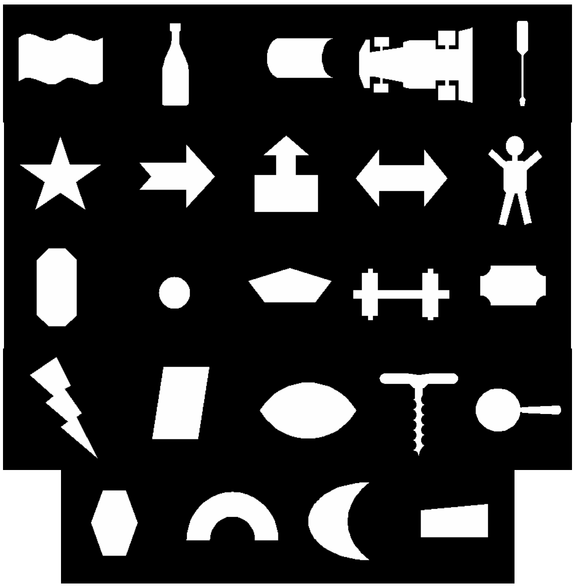

Figure 6: 24 object database

For the tests, an object is excluded from the database and the remaining objects are presented randomly to the user, in order to register grasping examples. After 40 examples (some objects are presented twice or even more than twice to the user) the decision trees are computed and the system has to synthesize the grasp of the excluded object.

The first experimental tests were performed only with those objects whose grasps are not operation dependant, and some results are shown on figure 7. For each object, there are three images: in the first image the valid contact points (output of step 1 of the algorithm) are marked on the contour of the object; in the second image the valid sets of contact points (sets of two points because a two jaw gripper is used) are marked: this is an intermediate result of step 3; and in the third image only the selected pair of contact points is shown: this is the final result of step 3 and –for a two jaw gripper- the final result of the algorithm.

The first experiments show a good behaviour of the system with previously unseen objects after only 40 grasp examples.

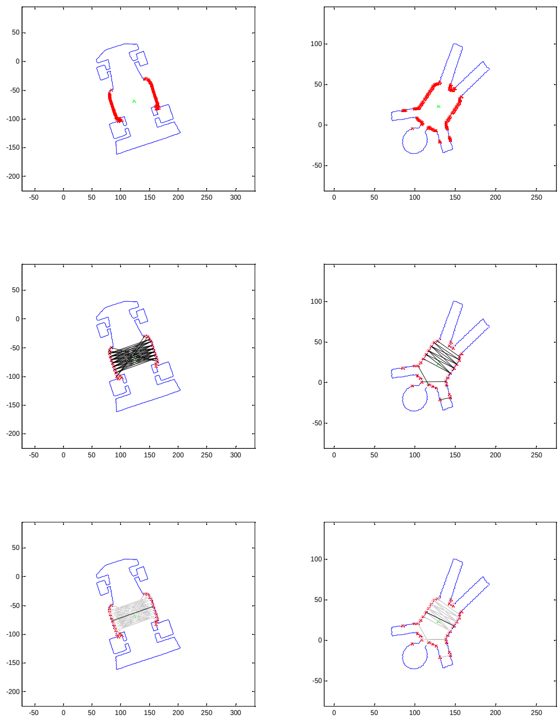

Figure 7: results with common objects

For the second set of experiments, all the objects in the database were used. That means that the user gave examples of merely geometrical grasps (those performed taking into account the geometry of the object) mixed with examples of operational grasps (those related to the operation to be performed with the object, e.g. a pan is grasped at its handle). After 40 examples, the decision trees were computed and the system was ready to synthesize grasps autonomously. The results with common, geometrical grasps were not affected to a high extent: the grasping points chosen autonomously were different to those of the first experiment but were still valid. Concerning the operational grasps, the results were computed in a slightly different way: in this case the object to be tested was included in the set given to the user to perform example grasps. When the program started working autonomously it became clear that the system had learned to perform operational grasps. Figure 8 shows the results with the corkscrew and the screwdriver as an example: it can be seen that both objects are grasped at their handles. Similar results are obtained with the other operational grasps: the bottle and the pan.

At present, the system is being installed on a physical experimental setup. For this purpose, a Mitsubishi RH-5AH SCARA robot equipped with a two jaw pneumatic gripper is being used. A top view stationary camera is in charge of acquiring the images of the objects, and a uniform background is used in order to avoid problems in contour extraction. Results with this experimental setup will be available soon.

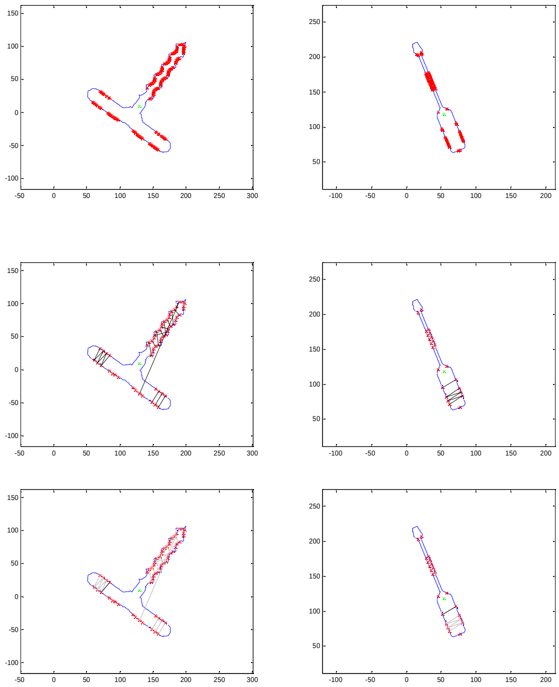

Figure 8: results with operational grasps

7 Conclusions

Robot grasp synthesis techniques are applicable for partial task automation in teleoperation and service robotics.

Learning based grasp synthesis outperforms non learning based methods as it allows taking into account the operation to be performed with the object to be grasped.

The four step algorithm proposed is general enough to work with simple or complex robot hands and to perform both 2D and 3D grasps. In this sense, it can be used as a general frame where each step can be modified depending on the requirements.

An algorithm like the one proposed can work in real time due to the reduction in computational complexity obtained by decoupling local and global characteristics of the grasps.

A combination of decision trees and nearest neighbour techniques has been selected for the learning based steps of the algorithm, because of its high descriptive power and good classification performance. Other classifiers could be used without modifying the structure of the algorithm.

Future work includes the improvement of the simulation environment in order to cope with more complex robot hands and the extension to 3D grasps, but the main structure of the algorithm will remain unchanged.

References

[1] L. Parker, J. Draper, "Robotics applications in maintenance and repair", *Handbook of industrial robotics 1023/1036*, Wiley Publishers, 1999.

[2] R. Aracil et al. "Advanced teleoperated system for live power line maintenance" *1st IFAC conference on telematics applications in automation and robotics*, Weingarten, 2001.

[3] L. Basañez, "Actualidad y perspectivas de la robótica", *Producción mecánica, 5,* Metalunivers, 2003.

[4] T. Fong, C. Thorpe, C. Baur, "Multi-robot remote driving with collaborative control", *IEEE Transactions on Industrial Electronics, 50(4)*, 2003.

[5] http://www.ifr.org/

[6] B. Graf, M. Hans, J. Kubacki, R.D. Schraft, "Robotic Home Assistant Care-O-Bot II", *Proc. of the 2nd Joint Meeting of the IEEE Eng. in Medicine and Biology Society and the Biomedical Eng. Society*, Houston, 2002.

[7] H. Kwee et al. "The Manus wheelchair-borne manipulator. System review and first results", *IARP 2nd Workshop on medical and healthcare robotics*, 1989.

[8] http://www.sony.net/SonyInfo/QRIO

[9] http://www.service-robots.org

[10] V.D. Nguyen, "The synthesis of stable force-closure grasps", *Technical report AI-TR-905,* MIT Artificial Intelligence Laboratory, 1986.

[11] E. Toth, "Stable object grasping with dextrous hand in three-dimension", *Periodica Polytechnica Ser. El. Eng.* vol. 43, No. 3, 1999, pp. 207-214.

[12] C. Ferrari, J. Canny, "Planning optimal grasps", *Proc. IEEE Conf. on Robotics and Automation*, Nice, 1992.

[13] J. Cornellá, R. Suárez, "On 2D 4-finger frictionless optimal grasps", *16th IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, 2003.

[14] N.S. Pollard, "Synthesizing grasps from generalized prototypes", *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, 1996.

[15] I. Kamon, T. Flash, S. Edelman, "Learning to grasp using visual information", *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, 1996, pp. 2470-2476.

[16] D. Schwammkrug, J. Walter, H. Ritter, "Rapid learning of robot grasping positions", *Proc. 7th Int. Symp. Intelligent Robotics*, 1999, pp. 149-155.

[17] C. Fernández, M.A. Vicente, C. Pérez, O. Reinoso. "Learning to grasp from examples in telerobotics", *Proc. 3rd Int. Conf. Artificial Intelligence and Applications*, Benalmádena, Spain, 2003.

[18] T.M. Mitchell, *Machine learning*, McGraw-Hill, 1997.

[19] D. Michie et al. *Machine learning, neural and statistical classification* , Ellis Horwood, 1994.

[20] J.R. Quinlan, *C4.5 : Programs for machine learning* , Morgan Kaufmann, 1993.