INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Robot Grasp Learning by Demonstration without Predefined Rules

Regular Paper

César Fernández, María Asunción Vicente, Ramón Pedro Ñeco and Rafael Puerto

Miguel Hernandez University, Elche, Spain
* Corresponding author E-mail: c.fernandez@umh.es

Received 21 Dec 2011; Accepted 19 Jan 2012

Abstract A learning-based approach to autonomous robot grasping is presented. Pattern recognition techniques are used to measure the similarity between a set of previously stored example grasps and all the possible candidate grasps for a new object. Two sets of features are defined in order to characterize grasps: *point* attributes describe the surroundings of a contact point; *point-set* attributes describe the relationship between the set of n contact points (assuming an n-fingered robot gripper is used). In the experiments performed, the nearest neighbour classifier outperforms other approaches like multilayer perceptrons, radial basis functions or decision trees, in terms of classification accuracy, while computational load is not excessive for a real time application (a grasp is fully synthesized in 0.2 seconds). The results obtained on a synthetic database show that the proposed system is able to imitate the grasping behaviour of the user (e.g. the system learns to grasp a mug by its handle). All the code has been made available for testing purposes.

Keywords robot learning, grasping, human imitation, nearest neighbour.

1. Introduction

Unlike traditional manufacturing applications, new applications of robots require them to grasp different objects and tools, some of them previously unknown. Service robotics [1] is one of these new fields of application, where robots coexist with humans, work in human environments and use tools that have been designed for humans.

The first task for an automatic grasp synthesis algorithm is the selection of the best contact points, i.e. those points on the surface of the object where the robot fingers are to be placed. Additionally, other problems like robot redundancy (multiple robot configurations can reach the same contact points) or trajectory selection have to be addressed, but such aspects are outside of the scope of the present paper. We will focus on contact point selection.

Our approach is different to that of most authors, since we consider grasp synthesis to be a pattern recognition problem. In our case, such a problem involves a classification step, where the goal is to classify all possible grasps as valid or invalid according to their similarity to a set of training examples, and a selection step, where the goal is to select, among all valid-classified grasps, the one

closer to a training example. No other grasp quality measure is considered but the similarity to training examples.

Most previous approaches for grasp synthesis can be roughly divided in two groups: those completely relying on predefined rules or heuristics and those making partial use of machine learning algorithms.

In the first group, some approaches try to find grasps fulfilling different criteria, like force and torque balance [2][3], force closure [4][5][6], form closure [7][8] or second-order immobility [9]. Roa [10] proposes an algorithm capable of determining contact regions fulfilling force closure, thus allowing for lack of precision in robot fingers. Goldberg [11] considers deformable parts and introduces *deform closure* grasps as those where positive work is needed to release the part. Other approaches try to maximize different quality functions: Cornella [12], Ferrari [13] and Mirtich [14] look for a grasp capable of resisting the maximum external force or torque; and Markenskoff [15] tries to minimize the force required in the robot fingers in order to hold a certain object. A different strategy is used by Cutkosky [16] and Pollard [17]; basically, predefined generalised grasps are adapted to each particular object to be grasped. A biologically inspired approach can be found in [18], where grasp synthesis is integrated with the incremental extraction of 3D information. A multi-resolution representation of the object allows the system to analyse in detail those object features more relevant for grasping. The system searches for possible graspable features, considering that the object is unknown, while a parallel search for previously executed grasps is carried out. Such parallelism resembles that of the dorsal and ventral streams in human grasping behaviour.

In the second group, Morales [19] proposes to use a multilayer perceptron or the nearest neighbour technique to infer the robustness of a grasp from previous experiments. Such robustness is used to select the best grasp among a set of different grasps synthesized using predefined rules (confluence of the surface normals at the contact points). The approach presented by Kamon [20] is based on an association of the contour of each object with the grasp to be performed. When an already known object is presented to the system, the previously stored grasp (specified as a set of parameters related to the object contour) is performed. When a new object is presented to the system, the grasp is synthesized according to some heuristic criteria, and its expected quality is inferred from the examples using the nearest neighbour technique. Platt [21] tackles the problem of grasp adjustment from an initial, approximately correct, grasp. He proposes different null space controllers which make use of the information provided by contact sensors to displace the robot fingers in order to reach a force closure configuration. In [22], Ekvall presents an approach based on the use of hidden Markov models to detect the kind of grasp being performed by the user in a teleoperation scenario (where there are several predefined grasps). Once the kind of grasp has been detected according to the initial sequence of movements performed by the user, the system is able to finish the task autonomously. A complete setup for robot grasping through human behaviour observation can be found in [23], where a humanoid robot is used to mimic human grasps. Both upper body movements and hand movements are captured (using computer vision) and mapped to the robot kinematics. Five different grasp types can be recognized and mimicked; however, only previously known objects can be grasped.

Even though learning techniques are sometimes used, none of the previous approaches is completely learning-based. All of them rely on predefined grasping rules or heuristics to synthesize the grasps, machine learning being added just to improve the behaviour of the system. We propose a fully learning-based system which tackles the grasp synthesis problem as a pattern recognition task.

To our knowledge, the only similar approach can be found in [24], where grasping points for novel objects are predicted from a database of synthetic examples. Such examples are 2D renderings obtained from 3D models of different objects with predefined grasping points. There are 2500 examples, and the grasping areas are described by a feature vector of dimension 459 (which includes both edge and texture information). When a new object has to be grasped, its image is divided in rectangular patches, and all the patches with similar features to those of one of the examples are considered correct grasping points.

2. Grasp description proposed

Grasp description is one of the main keys to success in learning-based grasp synthesis algorithms. The final goal is to establish a similarity measure in order to decide whether a new grasp is similar to the stored grasp examples or not.

Most previous research in robot grasp synthesis is based on the definition of quality measures; such quality measures are used to compare different grasps and to select the best one. Even though such approaches are not focused on pattern recognition, we have used their quality measures as a guideline in selecting the attributes for the learning-based approach proposed in this paper.
Some of the quality criteria used by previous authors (partly mentioned in section 1) are:

- Curvature of the object surface in the contact points. Gripper fingers are preferably placed on planar surfaces [2].

- Force and torque balance. The search is directed towards balanced grasps, capable of cancelling gravity or inertia forces. Some examples can be found in [2][3].
- Force closure or distribution of contact points capable of resisting arbitrary external forces acting on the object [4][6].
- Form closure and second-order immobility. Both quality criteria are related to the immobilization of the object, provided that the contact points are not displaced (high stiffness is assumed in the gripper fingers) [8][9][25].
- Maximum external force/torque resistible in any direction. There are different variations of such criteria, depending on how the force applied by the gripper fingers is limited; a detailed analysis can be found in [26]. Some grasp synthesis algorithms using this quality criteria are [12][13][14]. An approach providing closed-form expressions for the passive force closure set of grasps can be found in [27].

Some of the above mentioned quality criteria are related to points (they are measured at a single contact point, e.g. curvature) while others are related to sets of points (they depend on the relation between the contact points, e.g. force/form closure). The approach proposed in this paper is based on decoupling what we call *point* attributes and what we call *point-set* attributes. First, *point* attributes are measured in all the surface points in order to select those points valid for placing a robot finger on them, according to the training examples given in demonstration. Then, only those combinations of valid surface points are tested using the *point-set* attributes so as to select those combinations which are most similar to the training examples.

Briefly, a set of *point* attributes is needed, at first, in order to describe each candidate contact point on the surface of the object; then a set of *point-set* attributes is needed in order to describe each candidate set of n contact points (n being the number of gripper fingers).

2.1 Point attributes

These are the point attributes that we have selected for our system:

- Distance from the contact point to the centre of gravity of the object.
- Local curvature of the surface at the contact point (convexity).

Distance to the centre of gravity of the object allows the distinguishing of centred grasps from non-centred ones; while local curvature or convexity defines the kind of surface areas where the gripper fingers are best placed, these surface areas can be flat, convex or concave.

However, this information is clearly not enough for describing the training grasps performed by the user. For example, detecting whether the contact points are located preferably on handles requires additional data. One of the novelties presented in this paper is a simple, yet effective, method for describing contact points. The idea is to describe the shape surrounding a certain point as a set of convexities measured at different resolutions.

Object contours are sampled at five different resolutions. At each resolution, the convexity value gives different information about the shape of the contour in the surroundings of the contact points. Higher resolutions offer local information, while lower resolutions offer global information. The sequence of convexity values obtained may be representative of a particular shape. This is exactly what we need in order to measure similarity between contact points in two different grasps. Fig. 1 shows how convexity varies with resolution for three different contact points of three different objects. Fig. 2 compares the convexity values obtained for those contact points, and it becomes clear that such a measure is representative of the surroundings of the contact point.

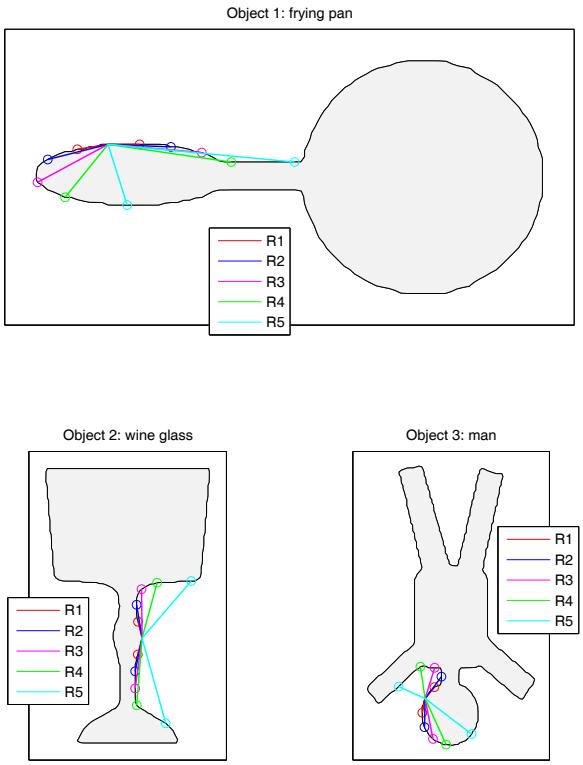

Figure 1. Multi-resolution convexity for three different contact points.

Regarding the resolutions chosen, we have decided to use 5 different resolutions, as a compromise between the amount of information and the complexity of the descriptor. Object contours are initially sampled so that there is a 1mm distance between two consecutive contact points (such a resolution can be easily obtained with a

video camera in a real application). Downsampling is easily performed just by discarding part of the original contour points and keeping some of them. In particular, the subsampling ratios used were 1/5, 1/10, 1/15, 1/20 and 1/30. Such resolutions have been shown to perform reasonably well with our database. Of course, using different resolutions (or even a different number of them) could also lead to good results.

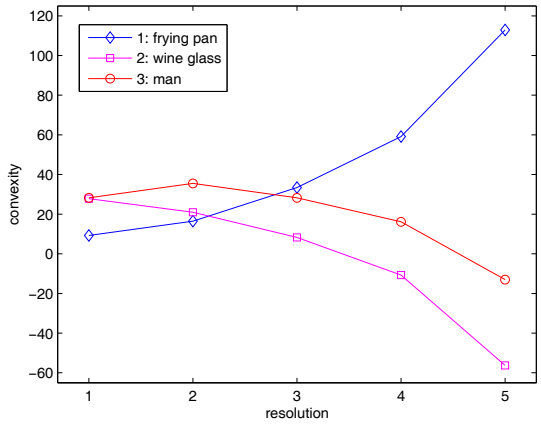

Figure 2. Multi-resolution contact point representation.

With regards to normalization, we have decided not to normalize the shapes (i.e. not to describe all the contours with the same number of points, independently of their size). Normalization discards size information which can be useful for describing a grasp, since it makes similar shaped contours look the same even if they have different sizes.

2.2 Point-set attributes

Point-set attributes are more difficult to choose. In our system, we have decided to define these attributes using a reference point. Such a reference point is the centre of gravity of the convex hull defined by all the contact points. We have considered two measures:

- Distance from the reference point to the centre of gravity of the object.
- Angle between the line directed to the reference point and the normal to the object surface at each contact point.

The distance to the centre of gravity of the object again allows the distinguishing of the centred grasp from non-centred ones, as Fig. 3 shows.

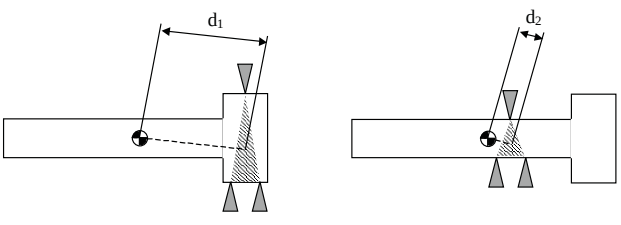

Figure 3. Centred vs. non-centred grasps.

The angular measure is related to the fulfilment of the force closure condition. When two-fingered grippers are used, a grasp fulfils force closure when the line joining both contact points lies within their friction cones [4]. Fig. 4 gives examples of grasps fulfilling or not fulfilling such a condition. Under such circumstances, the proposed attribute gives relevant information: as its value approaches zero, the grasp is more likely to fulfil the force closure condition.

When a three-fingered gripper is used, similar criteria can be established [5]: the grasp fulfils the force closure property if the friction cones of the contact points intersect each other (and the contact normals do not belong to the same half-plane). Fig. 5 provides an example. As the symmetry axis of the friction cones is collinear with the surface normal, the closer the angle between the normal and the line directed to the reference point gets to zero, the more likely it is that the force closure condition will be fulfilled.

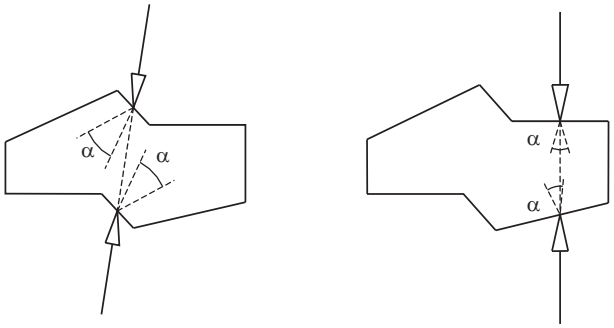

Figure 4. Force closure in two-fingered grasps. Left: non-force closure; right: force closure.

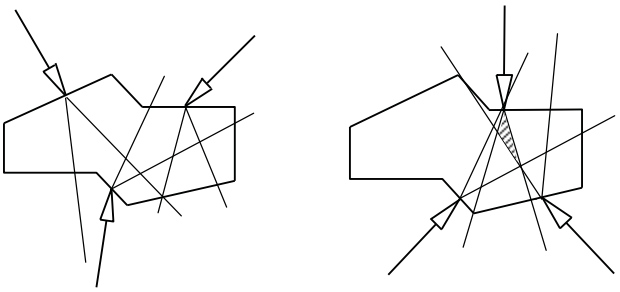

Figure 5. Force closure in three-fingered grasps. Left: non-force closure; right: force closure.

A similar approach to that of the *point* attributes has been adopted for *point-set* attributes. Instead of measuring the angles at only one resolution, they are measured at five different resolutions in order to describe in more detail the surroundings of the contact points. For reasons of coherence, the same five resolutions used for *point* attributes are also used for *point-set* attributes.

Arguably, these angular measures will not be related to force/form closure for low resolutions (since the contour shape is highly distorted). However, they will still be

representative of the kind of grasp being performed, so they may be useful in a pattern recognition scenario.

Let us consider an example. Given a two-fingered gripper - as with the grippers used in our experiments - a grasp is defined by a 23 dimensional vector, as Table 1 shows.

Cont. 1	Cont. 2	Ref. pt.	Norm. 1	Norm. 2
COG dist	COG dist	COG dist	-	-
Conv. R_1	Conv. R_1	-	Angle R_1	Angle R_1
Conv. R_1	Conv. R_2	-	Angle R_2	Angle R_2
Conv. R_3	Conv. R_3	-	Angle R_3	Angle R_3
Conv. R_4	Conv. R_4	-	Angle R_4	Angle R_4
Conv. R_5	Conv. R_5	-	Angle R_5	Angle R_5

Table 1. Two-fingered gripper grasp description.

In Table 1, *Cont. i* refers to the i^{th} contact point; *Ref. pt.* refers to the reference point; *Norm. i* refers to the normal at the i^{th} contact point; *Conv.* R_i refers to the convexity measured at the i^{th} resolution; and *Angle* R_i refers to the angle between the line directed from the reference point to the contact point and the normal at such a point, measured at the i^{th} resolution. Note that columns 1 and 2 contain the 12 *point* attributes; while columns 3, 4 and 5 contain the 11 *point-set* attributes.

3. Grasp synthesis algorithm

Our grasp synthesis algorithm can be described in terms of its off-line processes and its on-line processes.

3.1 Off-line processes

Once a set of grasp training examples has been inputted by the user, the following off-line processes are run:

1. *Point* and *point-set* attributes are extracted for each example grasp. The resulting 23 dimensional vectors (if two fingered grippers are used) are stored as correct grasp examples for the pattern recognition system.
2. For each example grasp, a random grasp (2 randomly selected grasping points) is generated, and its *point* and *point-set* attributes are also extracted. The resulting vectors are stored as incorrect grasp examples for the pattern recognition system.
3. Using the *point* attributes of correct and incorrect (randomly selected) grasps, a first model is created. Such a model can distinguish correct contact points from incorrect contact points.
4. Using the *point-set* attributes of correct and incorrect (randomly selected) grasps, a second model is created. Such a model can distinguish correct sets of contact points from incorrect sets.
5. For each grasp, all the attributes are stored for further use.

We have decided to include in our grasp example dataset the same number of correct and incorrect examples, in order not to obtain biased models. However, we do not expect the user to input wrong examples (it would not make sense), so these examples are generated randomly. There is an extremely low probability of generating a correct grasp randomly, but it may occur. Thus, the machine learning algorithms to be used should be able to cope with a small percentage of incorrectly labelled training examples.

3.2 On-line processes

Whenever a new grasp must be synthesized, the following on-line processes are run:

1. The contour of the object to be grasped is extracted (a good contrast with the background is assumed).
2. The contour is sampled so that there is a 1mm distance between two consecutive contact points (the same resolution used for the grasp examples).
3. The first model (*point* attributes) is used to classify all contour points as either valid or invalid for placing a gripper finger on them, according to the examples.
4. All possible sets of two valid contact points (if a two-fingered gripper is used) are analysed in order to detect which of them could be reachable by the robot gripper (more details about the algorithm used for such computation can be found in [28]).
5. The second model is used to classify all sets of two contact points as either valid or invalid grasps, according to the examples.
6. Among the correct sets of two contact points, the one which is most similar to one of the stored examples is selected as the best grasp for the object. Similarity is measured using Euclidean distance and considering all of the attributes (23 attributes for a two-fingered gripper).
7. Once the contact points have been chosen, a kinematic redundancy problem may arise, as more than one robot and gripper configuration may be able to reach such contact points. Even though this last step is out of the scope of the present paper, we propose to use an example-based approach, where the robot and gripper configuration more similar to those found in the grasp examples is selected (more details can be found in [29]).

Fig. 6 shows the on-line processes over an example object (the resolution has been lowered for a better visualization): First (upper left corner), all of the valid contact points are plotted over the object contour; second (upper right corner), the sets of valid contact points reachable by the robot gripper are outlined; third (lower left corner), only valid sets according to the examples are kept; and fourth (lower right corner), the final grasp is displayed.

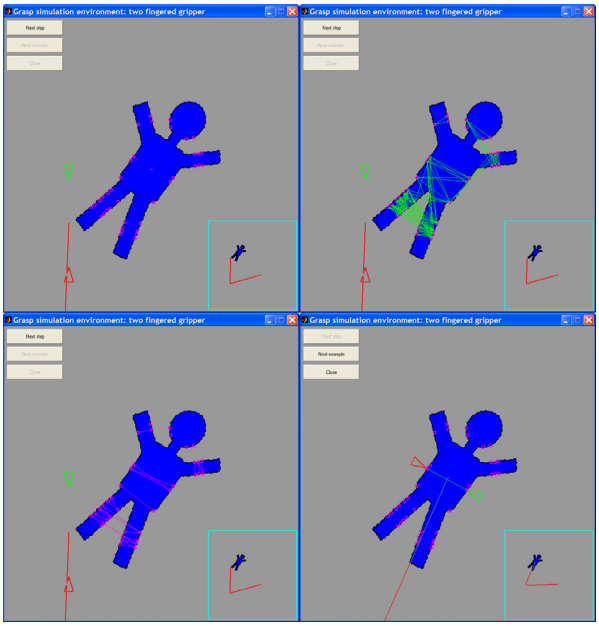

Figure 6. On-line processes. Left to right and top to bottom: valid contact point according to the examples; reachable contact sets; valid contact sets according to the examples; final grasp.

4. Model creation

Two of the off-line processes of our grasp synthesis algorithm require the creation of models: the third process, where *point* attributes are used to express rules that allow us to classify contact points; and the fourth process, where *point-set* attributes are used to express rules that allow us to classify sets of contact points. A comparison of different machine learning algorithms that could be used to build such models has been carried out. The following algorithms - or classifiers - have been tested:

- MLP: Multilayer perceptron.
- RBF: Radial basis functions.
- NN: Nearest neighbour.
- DT: Decision trees, specifically C4.5 algorithm [30].
- NB: Naive Bayes.
- RL: Rule lists, specifically PART algorithm [31].

Different criteria have been included in the comparison. First, some quantitative criteria: classification accuracy (measured by tenfold cross validation); the ratio between classification accuracy and the number of training examples used; and computing time, both off-line (training) and on-line (classification). In order to obtain these quantitative measures, 200 valid and 200 invalid grasps were generated using a simulation environment and two datasets were created: a dataset of *point* attributes corresponding to the third off-line process and a dataset of *point-set* attributes corresponding to the fourth off-line process (these datasets have been made available, in the Weka *.arff format, at [32]). Each classifier was tuned in order to produce the best possible results for each dataset. As the goal was to test the algorithms under conditions as similar as possible to those of a real setup, valid grasp examples were introduced by the user in demonstration, while invalid grasp examples were randomly selected.

Besides, three qualitative criteria have also been considered: readability of the results; sensitivity to modifications in the training parameters; and robustness to noisy training examples. The ideal algorithm should be readable (in order to allow for an easy interpretation of the model by a human); should have a low sensitivity to training parameters (in order to be easily implemented in different scenarios and to be reliable); and should be robust to noisy training examples.

Starting with the quantitative criteria, a previous tuning of each algorithm with the *point* attributes dataset led us to select the following configurations:

- MLP: 1 hidden layer with 9 neurons, learning rate set to 0.3, momentum set to 0.2, 500 epochs.
- RBF: 40 base functions.
- NN: 1 neighbour.
- DT: pruning confidence level set to 0.30.
- NB: probability density functions estimated as sums of Gaussians.
- RL: pruning confidence level set to 0.30.

Using such configurations, the comparative results obtained are described in Figs. 7 to 10.

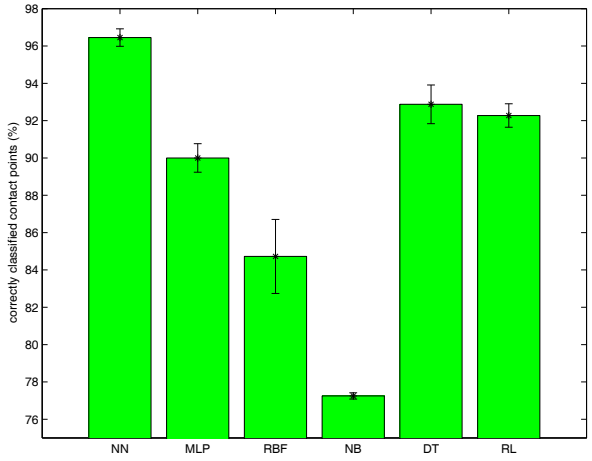

Figure 7. Cross validation results (*point* attributes).

Fig. 7 shows the results of a tenfold cross validation experiment repeated 10 times (average values and standard deviations are shown). It can be seen that the best performing algorithm for this application is NN (average 96.4% correct classification rate) followed by DT (92.9%) and RL (92.3%), these two methods giving approximately the same results. MLP, in fourth place, also gives good results (90.0%). Finally, RBF and NB do not seem to be suitable for this application.

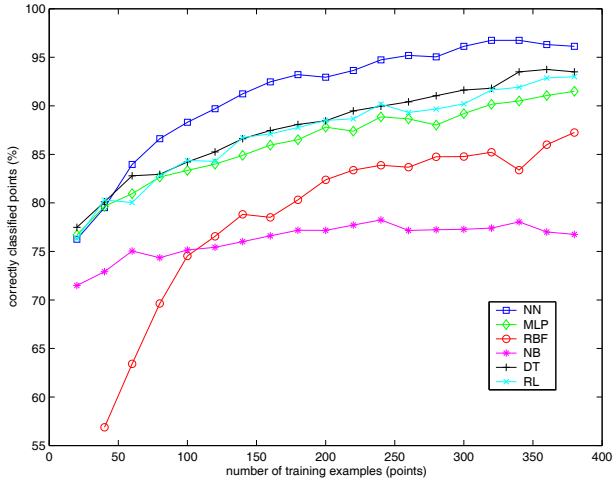

Figure 8. Classification accuracy vs. number of training examples (*point* attributes).

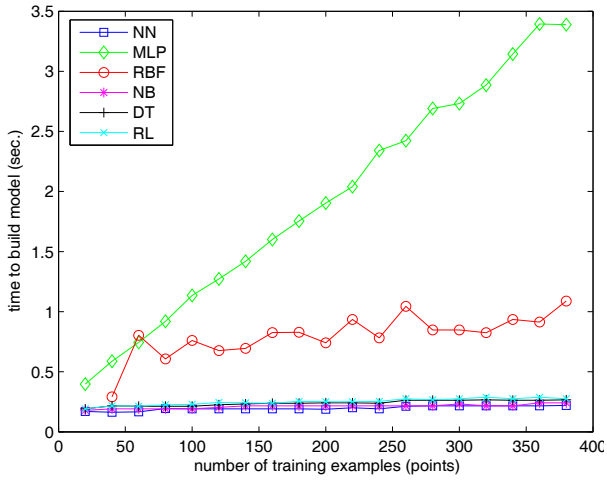

Figure 9. Off-line computing time.

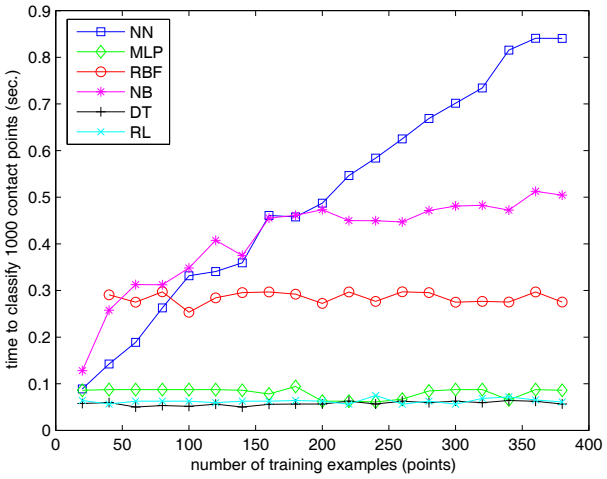

Figure 10. On-line computing time.

Fig. 8 shows the classification accuracy when the number of training examples varies from 20 to 380. The results obtained are similar to those of the cross validation experiment: NN is the best performing algorithm in almost all of the range of training examples, while DT, RL and MLP follow. In addition, this experiment allows us to conclude that the proposed grasp synthesis algorithm could work reasonably well with a small number of grasp examples: classification accuracy is around 80% when only 40 contact point examples are considered. As each grasp involves at least two contact points, this means that only 20 grasp examples are needed; besides, 50% of the grasp examples are randomly chosen (invalid grasps), so the user would only have to perform 10 correct grasps in demonstration mode and the system would have enough information to start working autonomously.

Concerning off-line computing times (measured on a standard 2.9GHz, 4GB RAM Intel Core i3 PC running Windows 7), Fig. 9 shows that the slowest algorithm is MLP, with a computing time approximately linear on the number of training examples and reaching 3.4 seconds for a dataset of 380 examples. The other algorithms are clearly faster and they are not so dependent on the number of examples. However, as training is performed off-line, these differences in computing times are not relevant for choosing among the candidate algorithms.

On-line or classification computing time is more relevant, because each time a new object has to be grasped, a huge number of points have to be classified. The results are shown by Fig. 10 as the time required to classify 1000 surface points. The fastest methods are MLP, DT and RL. RBF and NB are clearly slower, but still valid for our application (in the worst case, it took around 0.5 seconds to classify 1000 contact points). Finally, NN behaves as expected, with on-line computing times linear to the number of training examples. Thus, the feasibility of NN for our application depends on two factors:

- First, the number of training examples. According to Fig. 8, the best performances are obtained with around 300 training examples, but in a real application, the user is not expected to input more than 20 grasps (which gives 80 training examples, taking into account that every grasp involves two contact points and that for every correct grasp input by the user, a randomly selected grasp is added by the system).
- Second, the number of points to classify. Fig. 10 considers 1000 points, but the number of points will depend on the resolution chosen. We have decided to sample the object contour so that there is a 1mm distance between two consecutive points, which will give around 300 points for a standard object with a 30cm contour.

Even considering 200 training examples and 300 points, NN classification would take around 0.15 seconds, which is clearly acceptable for our application.

The same quantitative criteria, when measured over the *point-set* attributes dataset, gave slightly different results. First, the tuning of the algorithms was modified in order to obtain the best performances:

- MLP: 1 hidden layer with 7 neurons, learning rate set to 0.3, momentum set to 0.2, 500 epochs.
- RBF: 10 base functions.
- NN: 1 neighbour.
- DT: pruning confidence level set to 0.25.
- NB: probability density functions estimated as sums of Gaussians.
- RL: pruning confidence level set to 0.15.

Using such configurations, the comparative results obtained are described in Figs. 11 and 12 (off-line and on-line computing times are omitted as the results obtained are very similar to those shown for *point* attributes).

Concerning the tenfold cross validation experiment of Fig. 11, the results are similar to those obtained using the *point* attributes: NN is again the best performing algorithm (98.6% average), but MLP (98.1%) offers very similar results, outperforming RL (97.3%) and DT (96.8%); both of them offering again very similar results. NB gives slightly lower correct classification rates and, finally, RBF gives poor results with this dataset.

As Fig. 12 shows, the relative performance of the different algorithms is kept uniform in almost all of the range of training examples: NN is the best method followed by MLP, RL, DT, NB and RBF. However, NB is particularly well suited for applications where the number of training examples is very small (100 or below): in that range, NB offers results similar to those of NN and MLP.

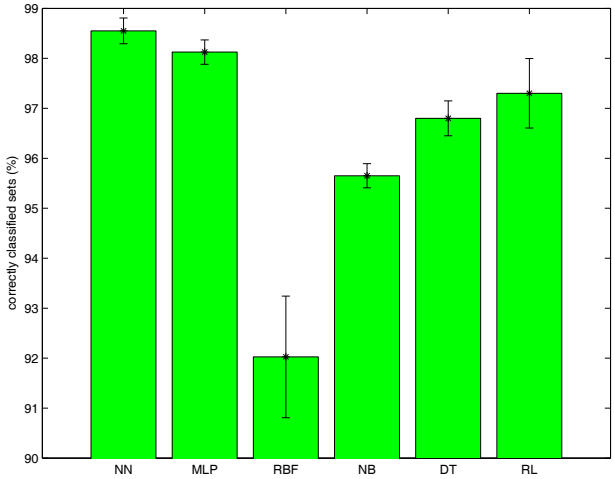

Figure 11. Cross validation results (*point-set* attributes).

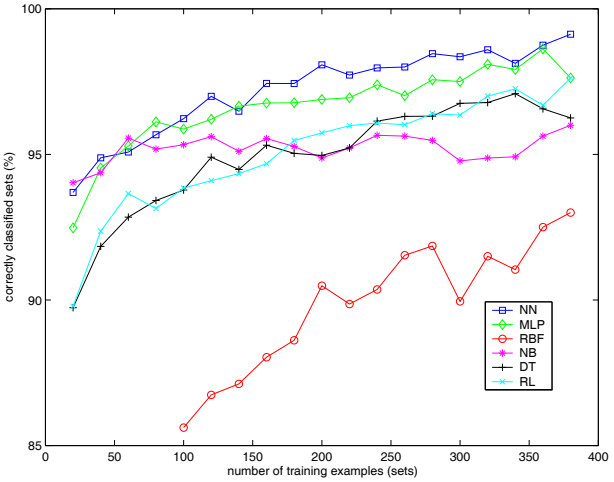

Figure 12. Classification accuracy vs. number of training examples (*point-set* attributes).

Globally, the percentage of correct classifications is even higher in this model (*point-set* attributes) than in the previous one. With only 40 grasp examples, the correct classification rate ranges from 92% to 95% (depending on the classifier used). These results, together with the results obtained for the previous model, confirm that the choice of attributes made in section 2 and the multi-resolution scheme adopted are adequate for representing the differences between valid and invalid grasps.

Apart from the quantitative tests, a qualitative comparison has also been performed in order to select the best classifier for the application. Table 2 shows the qualitative advantages of each method. As far as readability by a human is concerned, only DT and RL offer readable models. Concerning robustness to noisy training examples, all methods have been considered robust enough, except for local methods: NN and RBF (however, NN has performed extremely well with the datasets used for the experiments, which are inherently noisy: wrong grasps are randomly selected, so some of them may actually be correct grasps). Finally, concerning sensitivity to training parameters, only MLP and RBF seemed to be highly dependent on the parameterization (number of neurons in the hidden layer and number of base functions).

Method	Readable	Robust	Sensitive
NN	No	No	No
MLP	No	Yes	Yes
RBF	No	No	Yes
NB	No	Yes	No
DT	Yes	Yes	No
RL	Yes	Yes	No

Table 2. Qualitative comparison of machine-learning algorithms.

In brief, having a global look at both the quantitative and qualitative criteria, it must be concluded that the best

performing method in terms of classification accuracy is NN. Following in classification accuracy are MLP, DT and RL (depending on the experiment their relative performances are different, so there is not a clear winner).

Among these four methods, both NN and MLP lack readability; and MLP is too sensitive to modifications during parameter tuning (in particular, choosing the right number of neurons in the hidden layer is not straightforward). So, if we focus on classification accuracy, the best method would be NN, but if we also consider readability and other qualitative aspects, DT and RL may be more appropriate choices.

Readability could be an important advantage for a system such as the one proposed because it would allow the user to verify the results of the machine learning process. However, the sizes of the decision trees and rule lists obtained in our tests made them non-readable in practice. Fig. 13 shows the average number of nodes of the decision trees obtained in our tests (standard deviations and a point cloud representation are also included); and Fig. 14 shows an example decision tree. Similar results were obtained for the rule list algorithm; thus, both methods cannot be considered readable in practice, and they have no advantages over NN.

Regarding on-line computing time, although the nearest neighbour algorithm is the slowest one when the size of the training dataset increases, its speed in ordinary scenarios is adequate for our application. A standard grasp, which involves two classification steps (*point* attributes and *point-set* attributes) is synthesized in around 0.2 sec., which is much faster than the time required for a robot arm and hand to move to the desired location in most applications.

In conclusion, then, nearest neighbour is selected as the best machine-learning algorithm for our application.

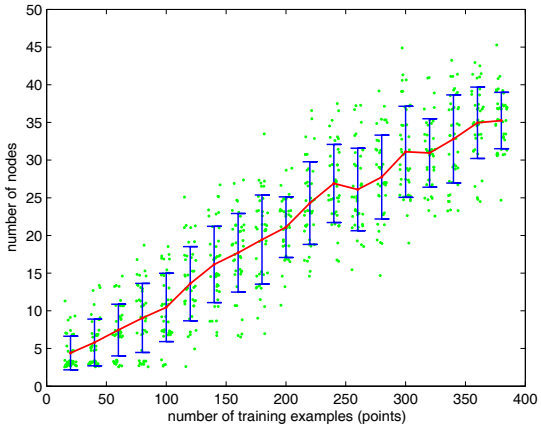

Figure 13. Decision tree size (number of nodes) vs. number of training examples.

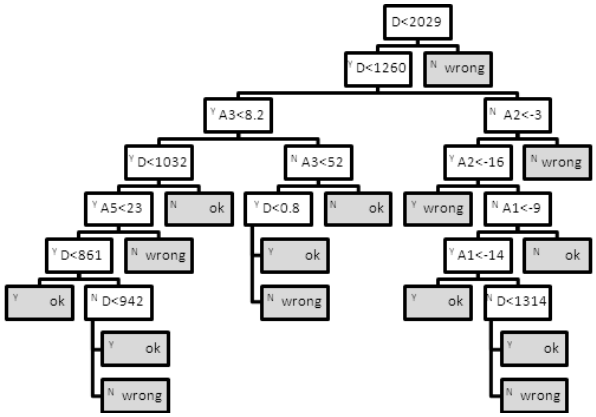

Figure 14. Example of decision tree (300 training examples, confidence level set to 0.30).

5. Experimental results

The goal of the algorithm proposed in this paper is not only to infer good grasps according to the shape of the objects (e.g. centred grasps or grasps fulfilling force/form closure) but to infer good grasps according to the operation to be performed with the object (e.g. objects with a handle should be grasped by it).

Taking the above considerations into account and in order to check the performance of our algorithm, four different tests have been performed:

5.1 First test: standard grasps

The training examples correspond to objects grasped close to the centre of gravity and using preferably planar surfaces. The training grasps performed by the user are shown with green arrows in Fig. 15. We followed a leave-one-out scheme, where every database object was autonomously grasped by the system using as training examples the grasps performed by the user for all other objects. The results are shown with red arrows in the same Fig. 15. The grasps can be considered appropriate in most cases (although appropriateness is a subjective measure).

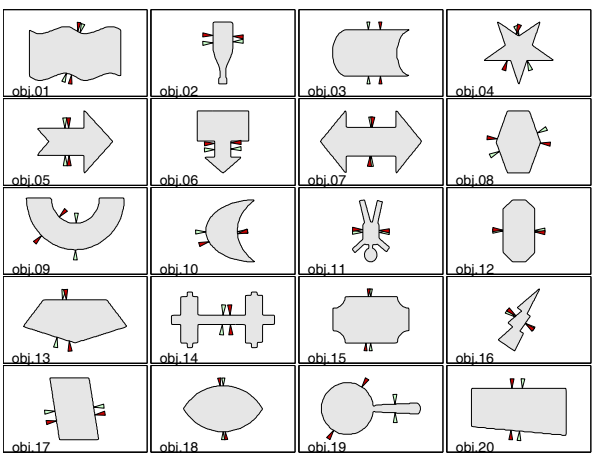

Figure 15. Results of the first test: standard grasps.

As expected, the autonomous grasps were not identical to the training grasps (as it has been said, the training data for the object to be grasped autonomously was kept apart), but the general behaviour (whenever possible, to grasp at planar or concave points and close to the centre of gravity) was correctly inferred. The only objects with grasps not completely fulfilling such behaviour were objects 8, 13 and 19 (one or both of the contact points fall on a convex area when there are other options); and object 9 (the grasp is correct, but it could be closer to the centre of gravity of the object). Anyway, it must be taken into account that the above mentioned grasping rules were not given explicitly: they were inferred from a small-sized set of training examples (19 examples) with no extra information.

5.2 Second test: particular grasping behaviours

The goal of the second test was to check whether the system imitates human behaviour. The training examples were grasps performed with a strong bias. In particular, contact points were placed close to the object vertexes, as Fig. 16 shows (green arrows). In order to check the ability of the system to imitate such behaviour, we followed a leave-one-out scheme similar to that of the first test.

The results can be seen in Fig. 16 (red arrows), where it becomes clear that the system has worked according to its expected behaviour. Once again, the autonomous grasps are obviously not identical to the training examples, but the main idea (whenever possible, grasp close to an object's vertex) has been correctly inferred. Of course, some of the grasps would not be reliable (e.g. objects 10 and 19): the goal of this second experiment was not to find good grasps, but to check to what extent the user behaviour was imitated (actually, some of the training grasps were also unreliable).

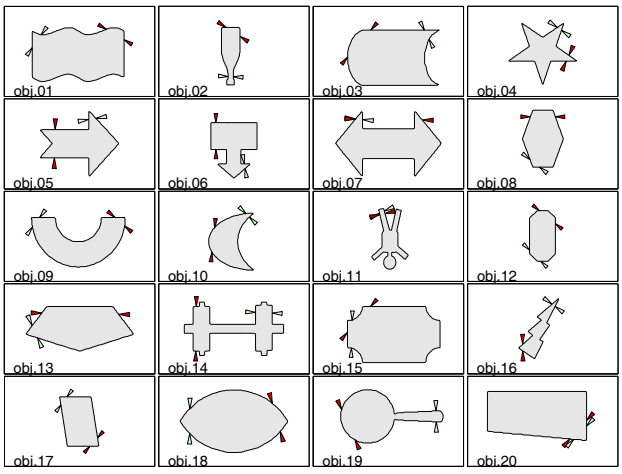

Figure 16. Results of the second test: particular grasping behaviours.

5.3 Third test: grasps related to operation

This is the most challenging test. A specific object database was designed, with objects belonging to three different categories: wine glasses, mugs and frying pans. The training examples try to tell the system that wine glasses must be grasped by their necks, while mugs and frying pans must be grasped by their handles. The whole database, the grasps performed by the user for every object, and the results obtained in the leave-one-out experiment are shown in Fig. 17. Each result (red arrows) corresponds to the grasp inferred by the system for that particular object, when all the other objects have been used as training examples (green arrows).

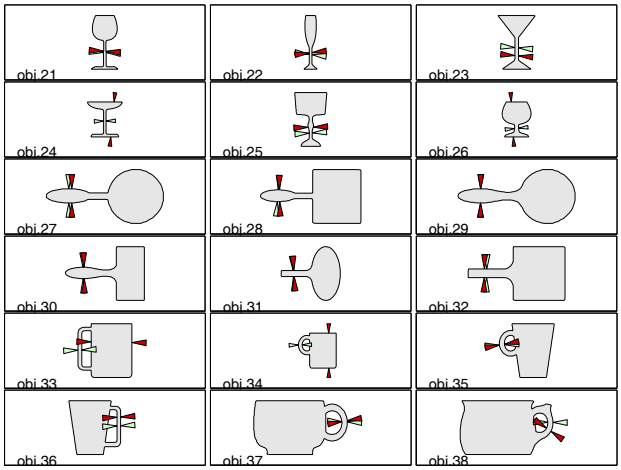

Figure 17. Results of the third test: grasps related to operation.

Even for a challenging test like this one, the results show that the system is able to infer correct rules without explicit information. All grasps must be considered adequate, except those of objects 24, 26, 33 and 34

However, the previous results of Fig. 17 show that the system is only able to infer specialized rules (rules for specific objects). We performed a further test in order to check whether our system was able to infer global rules: if the object has a handle, grasp it - preferably - by its handle; if it has not, grasp it - preferably - on a planar or convex area and close to its centre of gravity. We used as training examples all the examples of the first and the third test, where there is a mixture of different grasping strategies depending on the specific object. With this training data, we performed a leave-one-out experiment like those of previous tests. The results are shown in Fig. 18, and it becomes clear that the system has been capable of inferring global rules, valid for all kinds of objects.

Taking a close look at Fig. 18, the results must be considered quite good overall, although some grasps are not adequate. The wrong grasps are those involving object 9 (a similar error to that of the first test), object 19 (a frying pan must be grasped by its handle, but the system

has decided to perform a different grasp to that of all other frying pans, possibly due to the difference in size), objects 22 and 26 (wine glasses not grasped by their necks), and finally objects 33, 37 and 38 (mugs not grasped by their handles).

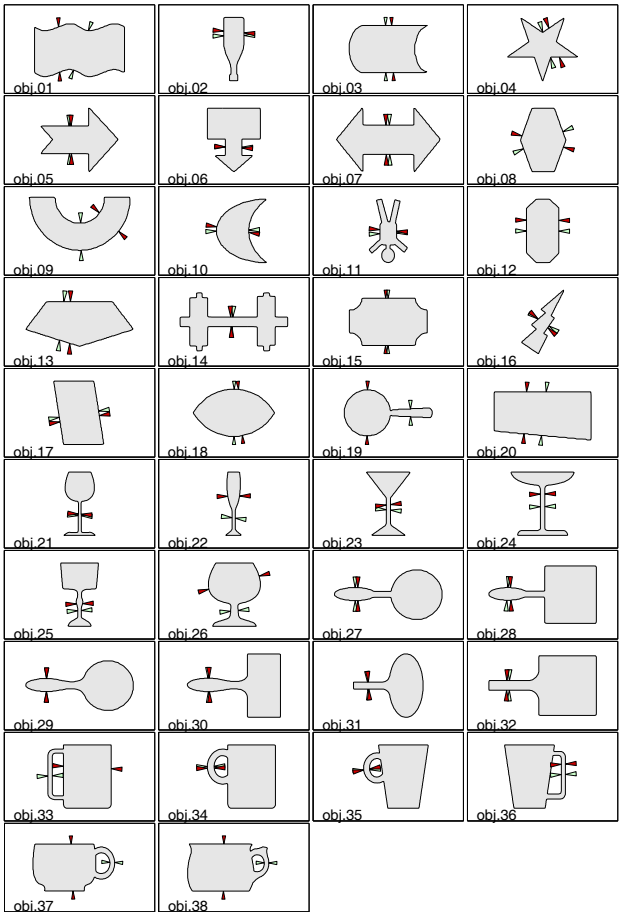

Figure 18. Combination of the first and third tests: mixed grasps.

5.4 Fourth test: supervision of results

According to the results of the previous tests, some of the objects are not grasped properly. One of the reasons is the way in which invalid grasps are obtained. Invalid contact points (see section 3) are selected randomly, so that in some cases they may actually be valid contact points. We will call these examples *false invalid examples*.

The supervision scenario we propose is simple: during autonomous behaviour the user simply needs to tell the system which of the synthesized grasps is not appropriate, and the system will add it to the training database as an invalid grasp, discarding one of the randomly selected invalid grasps previously in the database. Thus, there are two benefits:

- A new example of a wrong grasp (according to the user's opinion) is added to the training set.
- A randomly selected wrong grasp (which could be a false invalid grasp) is deleted from the training set.

Different tests carried out show that supervision helps in rapidly reducing the number of wrong grasps. As an example, in Fig. 19 the results of performing a single supervision step (one iteration) from the previous results of Fig. 17 are shown. Comparing the results, it can be seen that all the new grasps are slightly different to those obtained before the supervision step; the reason is that the grasping rules have been modified after introducing new examples to the database. With regard to the objects selected as wrong grasps in the supervision step (24, 26, 33 and 34), three of them (24, 26 and 34) are now grasped correctly, while object 33 still shows a wrong grasp. Further supervision steps can correct such behaviour.

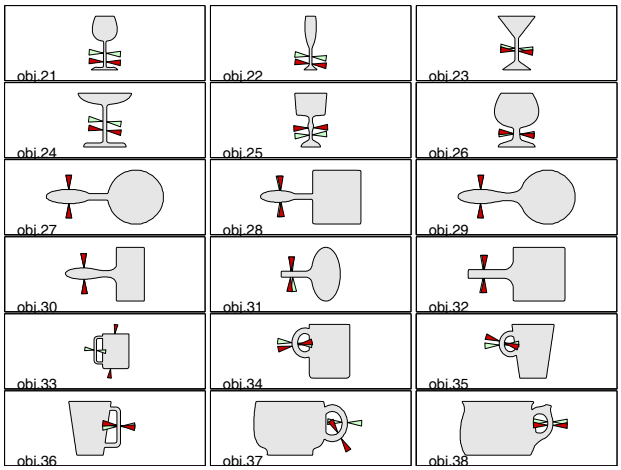

Figure 19. Results of fourth test: supervision loop.

All of the code which was developed as well as the object database and detailed instructions on how to use them (including a tutorial) have been made available at [32]. Thus, all the previous experiments can be easily verified and further tests can also be performed.

5.5 Experiments on a real setup

Our approach was also tested in a real scenario, using a Mitsubishi RH-5AH55 robot arm equipped with a two jaw parallel gripper and a video camera (see Fig. 20). The objects to be grasped were placed over a uniform background, where their contours were easily extracted by a region growing algorithm followed by a morphological filter.

The system was trained in simulation (using the standard grasp training set shown in section 5.1), and the inferred rules were used to grasp 12 different real objects. In our experiments, 92% of the grasps were successful (the object was picked, raised and displaced without falling), even though none of the 12 real objects were used during training. Some example videos are available at [32].

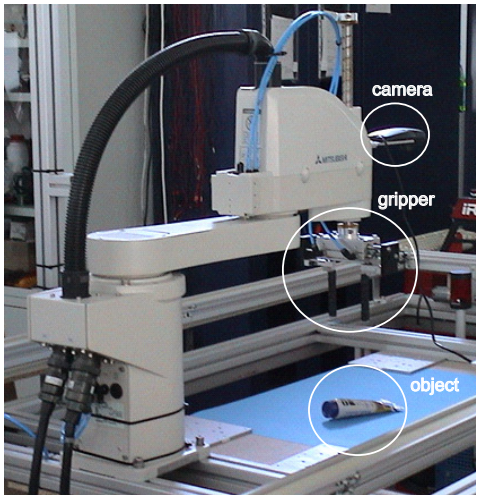

Figure 20. Experimental setup.

6. Conclusion

We have shown that it is possible to successfully synthesize robot grasps by means of pattern recognition techniques, without predefined rules, and relying only on the grasp examples given by the user.

The main advantage of our proposal is its ability to imitate user behaviour, as we have shown with different experiments.

Nearest neighbour seems to be the best performing pattern recognition technique for this particular application, although other options tested - such as multilayer perceptrons, decision trees and rule lists - also perform well.

Since grasp strategies are inferred from examples given by the user, there is no need to consider the mechanical properties of the object or gripper; the user takes this information into account when performing the grasp examples, and the system imitates such behaviour when grasping a new object.

Our approach is mainly focused on rigid objects; however, a learning-based algorithm like the one we propose can also cope with deformable objects, provided that enough grasp examples are supplied to the system. Tests with non-rigid objects will be considered in future work. Future work also involves the extension of our algorithm to 3D grasps. Such extension requires the definition of new grasp attributes.

7. Acknowledgments

The work presented in this paper has been funded by the Spanish *Ministerio de Ciencia e Innovación* through project TIN2010-17513.

8. References

[1] IEEE R.A.S., Technical Committee on Service robots. <http://www.service-robots.org>.

[2] M.A. Peshkin, A.C. Sanderson. *Reachable grasps on a polygon: the convex rope algorithm*. Technical report CMU-RI-TR-85-06, Robotics Institute, Carnegie Mellon University, 1985.

[3] N.S. Pollard, T. Lozano-Perez. Grasp stability and feasibility for an arm with an articulated hand. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1990, pp. 1581-1585.

[4] V.D. Nguyen. Constructing force-closure grasps. *Int. J. of Robotics Research*, vol. 7, no. 3, pp. 3-16, 1988.

[5] J. Ponce, B. Faverjon. On computing three finger force-closure grasp of polygonal objects. *IEEE Trans. on Robotics and Automation*, vol. 11, no. 6, pp. 868-881, 1995.

[6] J. Ponce, S. Sullivan, A. Sudsang, J.D. Boissonnat, J.P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Int. Journal of Robotics Research*, pp. 11-35, 1997.

[7] B. Mishra, J.T. Schwartz, M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, vol. 2, no. 4, pp. 541-558, 1987.

[8] B. Mishra, M. Teichmann. On immobility. *Laboratory Robotics and Automation*, vol. 4, pp. 145-153, 1992.

[9] A.F. Van der Stappen, C. Wentink, M.H Overmars. Computing immobilizing grasps of polygonal parts. *Int. Journal of Robotics Research*, vol. 19, no. 5, pp. 467-479, 2000.

[10] M.A. Roa, R. Suarez. Computation of independent contact regions for grasping 3D objects. *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 839-850, 2009.

[11] K. Goldberg, K. Gopalakrishnan, D-space and deform closure: a framework for holding deformable parts. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2004, vol. 1, pp. 345-350.

[12] J. Cornella, R. Suarez. On 2D 4-finger frictionless optimal grasps. *Proc. of the 16th IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 3680-3685.

[13] C. Ferrari, J. Canny. Planning optimal grasps. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 2290-2295.

[14] B. Mirtich, J. Canny. Easily computable optimum grasps in 2D and 3D. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 739-747.

[15] X. Markenscoff, C.H. Papadimitriou. Optimum grip of a polygon. *Int. Journal of Robotics Research*, vol 8, no. 2, pp. 17-29, 1989.

[16] M.R. Cutkosky, R.D. Howe. Human grasp choice and robotic grasp analysis. *Dextrous Robot Hands*, pp. 5-31, 1990.

[17] N.S. Pollard. Synthesizing grasps from generalized prototypes. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1996, vol. 3, pp. 2124-2130.

[18] G. Recatala, E. Chinellato, A.P. Del Pobil, Y. Mezouar, P. Martinet. Biologically-inspired 3D grasp synthesis based on visual exploration. *Autonomous Robots*, vol. 25, pp. 59-70, 2008.

[19] A. Morales, E. Chinellato, A.H. Fagg, A.P. del Pobil. Using experience for assessing grasp reliability. *Proc. of Int. Conf. on Humanoid Robots*, 2003, pp. 3423-3428.

[20] Y. Kamon, T. Flash, E. Edelman. Learning to grasp using visual information. *IEEE Trans. on Systems Man and Cybernetics*, vol. 28, pp. 266-276, 1998.

[21] R. Platt, A.H. Fagg, R.A Grupen. Null-space grasp control: theory and experiments. *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282-295, 2010.

[22] S. Ekvall, D. Kragic. Interactive grasp learning based on human demonstration. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 3519-3524.

[23] M. Do et al. Grasp recognition and mapping on humanoid robots. *Proc. of the 9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 465- 471.

[24] A. Saxena, J. Driemeyer, A.Y. Ng. Robotic grasping of novel objects using vision. *Int. Journal of Robotics Research*, vol. 27, no. 2, pp. 157-173, 2008.

[25] E. Rimon, J.A. Burdick. A configuration space analysis of bodies in contact--II. 2ND order mobility, *Mechanism and Machine Theory*, Elsevier, 1995, vol. 30, pp. 913-928.

[26] B. Mishra. Grasp metrics: optimality and complexity. *Algorithmic Foundations of Robotics*, pp. 137-166, 1995.

[27] A. Shapiro, E. Rimon, S. Shoval. On the passive force closure set of planar grasps and fixtures, The Int. Journal of Robotics Research, Multimedia Archives, 2010, 29, pp. 1435-1454.

[28] C. Fernandez, M.A. Vicente, O. Reinoso, L. Paya, R. Puerto. Grasp feasibility computation based on cascading filters. Application to a three fingered gripper. *Proc. of the 2nd ICINCO Conference*, 2005, pp. 181-188.

[29] C. Fernandez, M.A. Vicente, O. Reinoso, R. Aracil. Kinematic redundancy in robot grasp synthesis. An efficient tree-based representation. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1196-1201.

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

[31] E. Frank, I.H. Witten. Generating accurate rule sets without global optimization. *Proc. of the 15th Int. Conf. on Machine Learning*, 1998, pp. 144-151.

[32] Robot grasp learning in the LCSI webpage. <lcsi.umh.es/investigacion/robot-grasp-learning>.