2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Efficient and Effective Grasping of Novel Objects through Learning and Adapting a Knowledge Base

Noel Curtis and Jing Xiao, *Senior Member, IEEE*

***Abstract*— This paper introduces a new approach to establish a good grasp for a novel object quickly. A comprehensive knowledge base for grasping is learned that takes into account the geometrical and physical knowledge of grasping. To automate the learning process as much as possible, learning happens in a virtual environment. We used the GraspIt! [16] simulation environment with the Barrett hand for this work. As only approximate features of objects are used for training the grasping knowledge base (GKB), the knowledge gained is rather robust to object uncertainty. Based on the guidance of the GKB, a suitable grasp for a novel object can be found quickly. The newly gained grasping information of the new object can also be feedback to the GKB so that the knowledge base continues to improve as it is exposed to more grasping cases. The GKB serves as the "experience" of the robotic gripper to make grasping more and more skillful. We implemented the approach and tested it on a wide variety of objects. The results show the effectiveness of this approach to achieve quick and good grasps of novel objects.**

I. Introduction

The Grasp planning problem was initially addressed by geometric analysis of polyhedral objects and their wrench space, such as the grasping techniques presented by Ferrari and Canny [6], Mirtich and Canny [14] and by Wu and Hor [19]. Such approaches were not able to handle complex everyday (non-polyhedral) objects. Wrench space analysis formulated as a Linear Programming problem in the paper by Ding *et al.* [4] improved only the time complexity of such methods. Li and Sastry [11] tried to improve the grasp evaluation process in [6] by making it task oriented. Object models were first used in a grasp generation technique by Miller *et al.* [15]; their method modeled objects as a set of shape primitives and used preset pregrasp shapes to generate grasps for different objects. However, no knowledge was learned from grasping different objects, and physical characteristics of an object such as material type, weight, and distribution of mass were not taken into account in their work. Using object features to grasp novel objects was addressed by Saxena *et al.* [18], El-Khoury *et al.* [10], Kyota *et al.* [9] and also by Michel *et al.* [13]. The basic approach tries to extract a grasping point or a "handle" for an object by analyzing an image of the object. These methods require that such a "handle" be visible and reachable. Bowers and Lumia [1] used fuzzy logic to choose and validate grasps for simple three dimensional objects. Pelossof *et al.* [17] used a Support Vector Machine (SVM) [3] to generate grasps for the Barrett Hand. The SVM was trained via supervised learning to adapt parameters of the hand such as spread angle of the fingers and roll angle of the thumb to generate a grasp for super-ellipsoid objects. A technique using neural networks was proposed by Luo *et al.* [12], and a similar technique using reinforcement learning was proposed by Baier-Löwenstein and Zhang [2]. The above learning-based techniques focus on learning proper grasping of a specific object, and for a new object, learning has to be done again from scratch. Most of the existing methods also assume uniform mass distributions of objects.

This paper introduces a new approach to establish a good grasp for a novel, any-shape object quickly. Our approach accumulates knowledge gained from prior grasping to create a proper grasp for a new object efficiently. The accumulated geometrical and physical knowledge of grasping is saved in a knowledge base for grasping and object classification. This knowledge base continues to improve and adapt as more and more objects are grasped. Thus, through experience the grasping system becomes more and more efficient and robust, and it is able to grasp properly a wide variety of objects (including non-polyhedral objects, objects of different material, with non-uniform mass distribution so that the center of gravity is unknown, etc.) with large position and orientation uncertainty.

The following sections outline the basic assumptions made for the system, the creation and updating of the knowledge base, and the grasp generation process for novel objects. Finally some test results are provided involving example objects of various types to show the effectiveness of this approach.

II. Object Description and Basic Assumptions

We describe a graspable object in terms of rough shape, rough size, weight, and material type (e.g., wood, metal, etc.). The rough shape is estimated by fitting an oriented bounding box (OBB) [7] to the object, where length, width and height of the bounding box are used as parameters for the shape. They are also used to calculate the approximate volume of the object[1]. By taking the ratio between this volume and the maximum volume that can be grasped, a number between 0 and 1 is obtained, which is used to describe the *size* of the object.

Manuscript received February 22, 2008.
The authors are with the IMI Lab, Department of Computer Science, University of North Carolina – Charlotte, Charlotte, NC 28223, USA. E-mails: ncurtis@uncc.edu, xiao@uncc.edu.

[1] In the current implementation, we only used one OBB for each object, but that can be easily extended to using a combination of OBBs of different sizes for more accurate approximation of the shape of an object.

The reason of using shape and size approximations of an object is to facilitate object recognition through sensing in the presence of sensing uncertainty. It is assumed that the robot possesses a sensing system that can recognize a target object's rough shape, rough size, and rough configuration, as well as such information of nearby obstacles. This allows the robot to make an initial classification of the unknown object against known object classes in its knowledge base.

We use the Barrett Hand as the gripper in this work, which has three fingers and four degrees of freedom (DoF).

III. GRASPING KNOWLEDGE BASE STRUCTURE

The system uses a Grasping Knowledge Base (GKB) to store information about grasping different objects. The structure of the GKB is to classify objects and their grasps into different types. Each type is represented in a multi dimensional data structure. It has the following features: *rough shape, rough size, weight, material type*, and *a set of good representative grasps* consisting of grasping force and finger configurations with respect to a representative object's frame. The shape, size, and weight features are assumed to follow normal distributions so that each feature is represented by the mean value and the standard deviation of its normal distribution. Note that the rough shape feature follows a multivariate normal distribution (of three OBB variables: length, width, and height). Each representative grasp is described by a homogeneous transformation matrix for the pose of the gripper, the spread angle θ between two non-thumb fingers, the joint values of the three fingers, as well as a force matrix that shows forces applied by the fingers of the gripper. Table I shows some examples.

Table I: Representative objects of some object types in the GKB; finger force is from one representative grasp

Shape	Weight (g)	Size	Material	Finger Forces(N)
	50	0.4	wood	[3,3,4]
	150	0.7	metal	[4.7,4.7,5.3]
	70	0.5	plastic	[3.8,3.8,4.2]
	100	0.6	plastic	[4.3,4.3,5.6]
	500	0.8	glass	[9.1,9.1,9.8]
	30	0.3	wood	[2.2,2.2,3.6]
	600	0.8	glass	[11.2,11.2,11.7]

IV. KNOWLEDGE BASE INITIALIZATION

The initial creation of the knowledge base is a semi-automatic process. It involves the following.

A. *Creating an Initial Training Set*

A set of different representative objects for grasping are considered initially as seeds for different object types. These are objects of various simple shapes, sizes, weights, material types and poses. The rough shape, rough size, and weight information of a representative object is put into the GKB as the mean data for the corresponding features of the object type created. The material type of the representative object is used as the material type for the object type. While the rough shape and size information can be calculated automatically, the weight and material type is manually input to the system.

The representative grasps and knowledge about grasping for each type is the result of training as follows.

B. *Training to Grasp Representative Objects*

Training is done in a simulation environment provided by GraspIt! with a virtual Barrett hand. Virtual training is desirable before transferring the knowledge to a real-world robot hand. This is because it can not only automate the training process to make training with a very large set of various objects possible but also minimize damages to both the real-world objects and the robot hand.

For each object in the initial object set, which represents a different object type, a representative set of good grasps are needed. This grasp set must be sufficient to enable grasps from different possible directions. Such knowledge is necessary for grasping objects partially obstructed by obstacles.

i. Finding good grasps

The first thing is to decide the approach vectors of the end effector towards the object to be grasped. A set of approach vectors discretizing all possible grasping directions is created using the OBB box of the object. These approach vectors are considered at different positions intersecting the faces and edges of the bounding box around the object, see Figure 1 for an example. Once the set of approach vectors is generated, good grasps can be searched for systematically along each of these vectors, following Algorithm 1.

The function to evaluate the goodness of a grasp is computed via the grasp evaluation technique by Ferrari and Canny [6], which determines the worst disturbance wrench that can be resisted by a grasp of a unit strength, called the grasp wrench space projection *GWS(s)*.

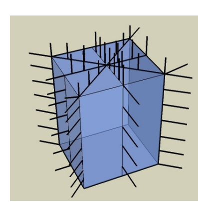


Figure 1. Approach vectors from the OBB box of an object.

ALGORITHM 1

```
for each approach vector do
    position the end effector along approach vector with a safe
    distance away from the object;
    initialize hand to be completely open;
    while palm has not contacted object do
        set spread angle to be the minimum;
        while maximum spread angle not reached do
            set minimum finger force;
            while maximum force not reached do
                increase all finger forces by Δf;
                execute grasp;
                if quality is greater than 0.4 then
                    record grasp;
                end if
            end while
            increase spread angle by Δd;
        end while
        set hand to be completely open;
        move along approach vector by +10 units
    end while
end for
```

ii. Clustering grasps and condensing grasp set

Running Algorithm 1 usually results in a large number (i.e., hundreds) of good grasps found for the training objects. To be efficient, we condense this set of grasps to a small subset of representative grasps sufficient to grasp the objects. Grasps in the original set are first grouped to clusters of similar grasps, and then a subset of similar grasps is selected from each of these clusters.

The *K*-Mediod algorithm [8] is used to cluster similar grasps together. It is use to hard partition the grasps based on different values of *K*, and the cluster centers are data points. The chosen cluster center is the nearest data point to the mean of the cluster. *K* is decided by evaluating grasp clusters for different values of *K*. Values of *K* ranging from 2 to 6 are tested for optimality using the *Partition Index* and *Separation Index*.

Partition Index (SC) is the ratio of the sum of compactness and separation of the clusters. It is the sum of individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster. *SC* is useful when comparing deferent partitions having equal number of clusters. A lower value of *SC* indicates a better partition. Let c be the number of clusters, v_i be the ith cluster center, N_i be the number of data elements in the ith cluster, x be a data element, and μ_{ij} be the membership of data point j in cluster i. *SC* is expressed in the following formula.

$$SC(c) = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N_i} \mu_{ij} \|x_i - v_i\|^2}{N_i \sum_{k=1}^{c} \|v_k - v_i\|^2}$$

Separation Index (S), on the other hand, uses a minimum-distance separation for partition validity. *S* is expressed below, where N_{min} is the minimum number of elements in a cluster, and d_{min} is the minimum distance between cluster centroids ($d_{min} = min_{i,j}\| v_i - v_j\|$).

$$S(c) = \frac{\sum_{i=1}^{c} \sum_{j=1}^{N_i} (\mu_{ij})^2 \|x_j - v_j\|^2}{N_{\min} (d_{\min})^2}$$

Note that when the results of the cluster validity measures for different *K*'s are close to one another, we choose the result with a smaller *K* (i.e., smaller number of clusters).

To select a subset of grasps, we pick *M* most directionally different grasps from each of the *K* clusters. Specifically, for each grasp in a cluster, we take the mean squared error (MSE) between one grasp configuration and all other grasps in the cluster, and then pick the *M* grasps with the largest MSE. The total number of grasps selected from the *K* clusters is thus *KM*. Our experience shows that it is usually sufficient to have *KM* in the range of 10 to 15 good grasps for this condensed set of representative good grasps.

The generated representative good grasps are stored in the GKB relative to the object frame as the *pregrasp set*.

V. GRASPING NEW OBJECTS

The following section explains the process of how grasps are generated for novel objects. The grasping process consists of two steps, as described below.

A. Object Classification

A novel object is presented to the grasping system with the following information (which can be from sensing): a rough shape and a rough size. After the novel object is presented, the system first tries to classify it against the object types in the GKB. The classification is done using the Nearest Neighbor algorithm [5]. First, the object shape is checked, and then the object size is checked, in order to identify the closest object type. Next, find the most suitable grasp for the new object taking advantage of the pregrasp set provided in the found object type. This leads to the following step.

B. Grasp Selection and Local Adjustment

Since the found object type in the GKB is most similar to the new object in appearance, it is reasonable to hope that certain small adjustment of a grasp in the pregrasp set of that object type may lead to a suitable grasp for the new object. Figure 2 shows an example, where a pregrasp for a cylinder is adjusted to make a good grasp of a mug of the same material.

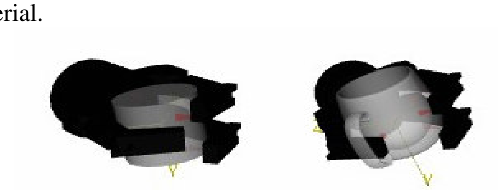


Figure 2. A grasp generated for a cylinder can be adjusted and used to grasp a mug.

From the pregrasp set, the system first selects the grasps that are not blocked by obstacles by checking if the approach vectors of these grasps will lead to collision of the gripper with the obstacles or not. The grasp with the highest *GWS(s)* score is selected from this sub set. The chosen grasp can now be tried on the new object and adjusted if needed.

To adjust the grasping force, if the new object's weight and material type are not known, they can be inferred through a look-up table that associate grasping forces for successful grasps to different weight scales and material types in the GKB. This look-up table can be built in the training process provided that the training set represents a large variation of weights and material types.

In case the above adjustment does not provide a good grasp, local search (such as hill climbing) can be used to search for a good grasp starting from the collision-free pregrasp by small adjustment of the spread angle, the force, and the approach vector.

If a good grasp can still not be found, and if the circumstance allows (e.g., there is no hurry for the robot to move on to a different task, or the new object is meant to be used as a training object), the object can be used to create a new object type in the GKB, as described in the following section.

VI. UPDATING THE KNOWLEDGE BASE

The GKB can be updated each time a new object is encountered for grasping. There are two kinds of updates: (1) update the information of an existing object type in the GKB, and (2) create a new object type.

The first kind of update is possible if a successful grasp is found for the new object by local adjustments based on the pregrasp set of the closest object type in the GKB. When the new object is considered for an existing object type, first the standard deviation σ of every feature distribution for that type is recalculated, and if every updated σ value is still smaller than a certain threshold by including the new object, the object is classified as belonging to that type. Otherwise, a new object type has to be created. If the object is classified as belonging to an existing type, its rough shape, rough size, and weight (if known) are averaged into the mean size, shape and weight of the object type in addition to the updated standard deviations. This works as learning the feature distributions of the object type as new objects belonging to the type are encountered. The good grasps generated for the new object can be used in updating the pregrasp set of that object type by modifying the grasp clustering and condensing results.

When a new object is classified into a new object type, as mentioned above, the new type is created by using the feature that distinguishes it the most from the existing type closest to the object. The search for grasps (section IV.B) will have to be redone. Note that the GKB grows much slower than the number of new objects it encounters, because the data of new objects that match an existing object type are incorporated in the type information and do not really grow the GKB. Only the addition of new object types makes the GKB larger.

VII. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The system was implemented using the C++ programming language along with a MATLAB interface and the GraspIt! simulator. The robot consists of the PUMA arm as the manipulator and the Barrett hand as the gripper. The system was trained using a set of varied representative objects and then tested with the task of generating grasps on a number of novel objects.

A. *Training Set*

The training set consists of primitive object shapes, and for each shape, a number of objects of varied other characteristics (i.e., size, weight, material type). Figure 3 shows some example objects from the training set.

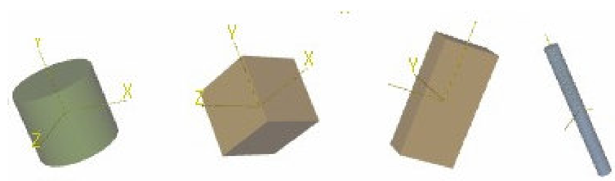


Figure 3. Example objects from the training set. From left to right: Metal Cylinder: 150g, Wood Cube: 300g, Glass Box: 600g, Plastic Rod: 50g

Table II shows some example object types generated for the GKB, where the mean value for the normal distribution of each feature is obtained based on the objects used in training, and the standard deviation for each distribution is pre-decided, proportional to the mean value. Specifically, for each parameter, a range of change is decided empirically, and then, 10 samples are randomly chosen within the range, and the standard deviation is computed from the samples. Note that *size* is a number between 0 and 1 as described in Section II.

Table II: Example GKB Object Types
(1. plastic, 2. wood, 3. metal, 4. glass)

Object Type Index	Shape(Bounded Box) l: length, w: width ,h: height (inches)	Size	Weight(g)
1	$\mu_l=2$ $\mu_w=1$ $\mu_h=5$ $\sigma_l=0.70$ $\sigma_w=0.40$ $\sigma_h=0.70$	μ=0.5 σ= 0.081	μ= 152 σ= 19.63
2	$\mu_l=2$ $\mu_w=2$ $\mu_h=7$ $\sigma_l=0.70$ $\sigma_w=0.70$ $\sigma_h=0.70$	μ=0.8 σ= 0.081	μ= 407 σ= 21.36
3	$\mu_l=3$ $\mu_w=3$ $\mu_h=4$ $\sigma_l=0.70$ $\sigma_w=0.70$ $\sigma_h=0.70$	μ=0.9 σ= 0.081	μ= 190 σ= 11.83
4	$\mu_l=2$ $\mu_w=2$ $\mu_h=3$ $\sigma_l=0.70$ $\sigma_w=0.40$ $\sigma_h=0.70$	μ=0.6 σ= 0.081	μ= 260 σ= 14.14

Table III shows the number of good grasps found for each object type in Table II.

Table III: Number of Good Grasps Found

Object Type	1	2	3	4
Number of Good Grasps	120	142	98	157

B. *Grasping Novel Objects*

Once a number of object types were established to form the GKB, we next tested how our system performed in generating grasps on a set of novel objects. Example objects from the test set are shown in Figure 4.

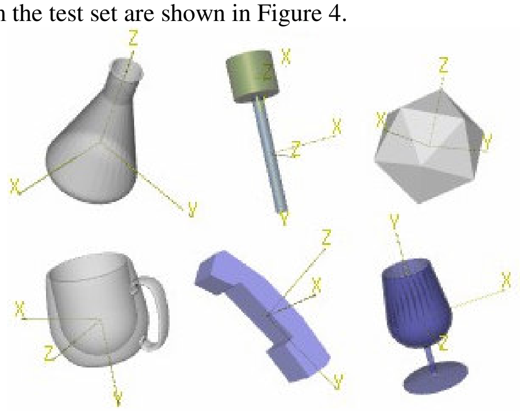


Figure 4. Left to right: flask, hammer, icosahedron, mug, phone, wine glass

Table IV compares a number of novel objects vs. the corresponding object types as well as a set of corresponding training objects in the GKB. The detailed information of those object types are already shown in table II. This table compares the time to find a good grasp for a training object from scratch vs. the time to find a good grasp for a novel object based on the knowledge of the object type in GKB obtained from the training objects. Clearly, finding a good grasp for the novel object took much less time by taking advantage of the knowledge of the object type. This shows the effectiveness of our GKB-based approach. Also shown is the corresponding number of adjustment steps needed to generate a successful grasp for each novel object. These local adjustment steps reflects that each object type information shown is obtained from only one training object, which is different from the corresponding novel object.

Table V shows two new object types generated based on the novel objects icosahedron and wine glass.

Table IV: Object Types and Training Objects vs. Novel Objects

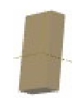
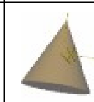
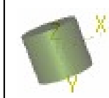
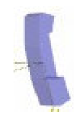
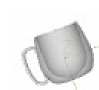
Object Type	1	2	3	4
Training Object				
Features	S: 0.5 W: 152g M: Plastic	S: 0.8 W: 407g M: Wood	S: 0.9 W: 190g M: Metal	S: 0.6 W: 260g M: Glass
Mean time to generate first good grasp(sec)	1.54	1.05	1.89	1.48
Training time(sec)	128	73	112	146
Novel Object				
Features	S: 0.4 W: 121g M: Plastic	S: 0.9 W: 360g M: Wood	S: 0.8 W: 180g M: Metal	S: 0.5 W: 250g M: Glass
Grasping Time(sec)	0.74	1.03	0.36	0.46
Adjustment Steps	14	23	9	12

Table V: New Object Types

Object Type Index	Shape(Bounded Box) l: length, w: width ,h: height (inches)	Size	Weight(g)
5	$\mu_l=2$ $\mu_w=2$ $\mu_h=2$ $\sigma_l=0.70$ $\sigma_w=0.40$ $\sigma_h=0.70$	$\mu=0.2$ $\sigma=0.081$	$\mu=72$ $\sigma=17.54$
6	$\mu_l=3$ $\mu_w=3$ $\mu_h=4$ $\sigma_l=0.70$ $\sigma_w=0.70$ $\sigma_h=0.70$	$\mu=0.5$ $\sigma=0.081$	$\mu=131$ $\sigma=21.36$

Figures 5 and 6 show grasps generated in the presence of obstacles, using the strategy as described in section V.B.

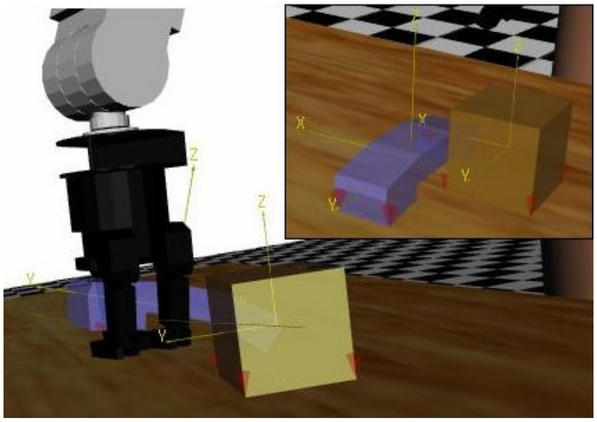


Figure 5. Upper right: arrangement of the phone and the obstacle (cube). Lower left: grasp generated for the phone. Grasping time: 1.73 sec, Number of adjustment steps: 163

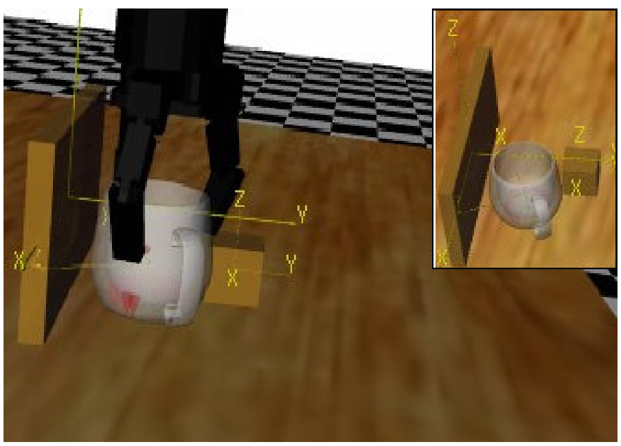


Figure 6. Upper right: arrangement of the mug and the obstacle (cube). Lower left: grasp generated for the mug. Grasping time: 3.03 sec, Number of adjustment steps: 286

VIII. CONCLUSIONS

This paper presents an effective method that uses learned grasping knowledge on a set of representative object types to greatly speed up the process of finding successful grasps for novel objects. It allows continuous accumulation and adaptation of the grasping knowledge base (GKB) as more and more objects are encountered so that the system becomes more and more skillful. Further work includes training more objects and more types of objects (such as objects of mixed material types) to improve the GKB. Real world testing involving a real robot hand and real objects is also necessary. This approach can be coupled with an object recognition system to form a robust grasp generation system that works efficiently on a wide range of novel objects.

REFERENCES

[1] D. Bowers and R. Lumia, Manipulation of Unmodeled Objects Using Intelligent Grasping Schemes. *IEEE Transactions on Fuzzy Systems*, pp. 320-330, June 2003.

[2] T. Baier-Löwenstein and J. Zhang, Learning to Grasp Everyday Objects using Reinforcement-Learning with Automatic Value Cut-Off. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1551-1556, San Diego, CA, USA, Oct. 29 - Nov 2, 2007.

[3] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines. *Cambridge University Press*, 2000.

[4] D. Ding, Y.-H. Liu, and S. Wang. Computing 3-D optimal form closure grasps. *Proc. of the 2000 IEEE International Conference on Robotics and Automation*, pp. 573–3578, San Francisco, CA, April 2000.

[5] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis. Wiley, 1973.

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

[7] Gottschalk *et al.* OBBTree: A Hierarchical Structure for Rapid Interference Detection. *Proceedings of SIGGRAPH Conference,* pp. 171-180, 1996.

[8] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics,* John Wiley and Sons, 1990.

[9] F. Kyota, T. Watabe, S. Saito, and M. Nakajima, Detection and Evaluation of Grasping Positions for Autonomous Agents. *Proceedings of the IEEE International Conference on Cyberworlds*, pp. 8-12, 2005

[10] S. El-Khoury, A. Sahbani and V. Perdereau. Learning the Natural Grasping Component of an Unknown Object. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2957 - 2962, San Diego, CA, USA, Oct 29 - Nov 2, 2007.

[11] Z. Li and S. Sastry, Task-Oriented Optimal grasping by Multifingered Robot Hands. *IEEE Journal of Robotics and Automation*, pp. 32-44, Vol 4. No. 1, February 1988.

[12] M. Luo, W. Lu, B. Sun, and T. Mei. Autonomous Grasping of a Space Robot Multisensory Gripper. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1014-1019, October 9 - 15, 2006, Beijing, China.

[13] 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,* pp. 3130 – 3135, Genoa, Italy, 2004.

[14] B. Mirtich and J. Canny, Easily Computable Optimum Grasps in 2D and 3D. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation,* pp. 739 - 747, 1994.

[15] T. Miller, S. Knoop, P. K. Allen, and H. I. Christensen, Automatic Grasp Planning Using Shape Primitives. *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1824–1829, 2003.

[16] T. Miller and P. K. Allen, GraspIt!: A versatile simulator for grasping analysis. *Proc. of the ASME Dynamic Systems and Control Division*, volume 2, pp. 1251–1258, Orlando, FL, 2000.

[17] R. Pelossof, A. Miller, P. Allen, and T. Jebara, An SVM Learning Approach to Robotic Grasping. *Proceedings of the IEEE International Conference on Robotics & Automation,* pp. 3512 - 3518, April 2004.

[18] A. Saxena, J. Driemeyer, and A. Y. Ng, Robotic Grasping of Novel Objects, *International Symposium on Experimental Robotics*, 2006.

[19] S. Wu and M. Hor, Efficient and Robust Algorithm for 3D Grasp Construction. *Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 218 – 223, July 29-August 1, 2001.