Visual Guided Grasping of Aggregates using Self-Valuing Learning

Bernd Rössler, Jianwei Zhang
{broessle, zhang}@techfak.uni-bielefeld.de
Faculty of Technology, University of Bielefeld,
33501 Bielefeld, Germany

Alois Knoll
knoll@informatik.tu-muenchen.de
Technical University of Munich,
81667 Munich, Germany

Abstract

We present a self-valuing learning technique which is capable of learning how to grasp unfamiliar objects and generalize the learned abilities. The learning system consists of two learners which distinguish between local and global grasping criteria. The local criteria are not object specific while the global criteria cover physical properties of each object. The system is self-valuing, i.e. it rates its actions by evaluating sensory information and the usage of image processing techniques. An experimental setup consisting of a PUMA-260 manipulator, equipped with a hand-camera and a force/torque sensor, was used to test this scheme. The system has shown the ability to grasp a wide range of objects and to apply previously learned knowledge to new objects.

1 Introduction

In a wide range of robotic systems grasping is a basic skill that is crucial to manipulation tasks and interaction with the environment. In most industrial applications the problem of grasping is solved via *teaching-by-doing* or static programs. However, sensor-based motions like *visual servoing* are rarely implemented in these industrial robotic systems. But when thinking of recent research fields, e.g. service robots or humanoids, aspects of sensor based grasping will play a very important role. New techniques must be developed for the robots to operate in uncharted and unknown territories. They should consider elements of human learning abilities when constructing a robotic grasping system. Such an approach is presented in this paper.

2 Related Research

A lot of work has been done in the field of robot grasping. [1] gives a brief overview of the field over the last two decades. Most works deal with analytical approaches that try to compute optimal grips according to special heuristics (e.g. in [2] and [3]). In these cases one has either a fully specified model of the object and its mass distribution or one has to use the center of area of the object, extracted via image processing, to "approximate" the real center of gravity. The first case is very difficult to obtain via external sensors and without any previous knowledge. One would have to gain a complete 3D representation of the object via image processing and additionally try to examine things like the material of the object. However, a hidden internal inhomogeneous mass distribution can never be found with such an approach. The latter case of using the center of the object's area is certainly only a kind of approximation. This works fine if the center of gravity coincides with the object's center of area, but this approach cannot deal with inhomogeneity, too. However, relatively few efforts handle the problem of learning how to grasp. In [4] a system is presented that learns how to grasp objects with a parallel-jaw gripper. Two main subproblems are learned: to choose grasping points and to predict the quality of a given grasp. The disadvantage of this system is that only *local* criteria are used to store grasping configurations. Without *global* criteria it is for example impossible to learn how to grasp an object which center of gravity does not coincide with the center of its image area. Without self-valuing learning techniques it is not possible to handle the real physical properties of an object. [5] presented a learning system for visual guided grasping, constructed of two learners. This system is not self-valuing, i.e. the optimal grasp point has to be given initially to the learner. Therefore, the two learners are also not generalizable to new objects. In [6] an uncalibrated vision-guided system was developed for manipulating objects that may be placed anywhere in the robot's 3-D workspace even though not visible in the initial fields of view of the cameras.

3 Learning Scheme

To construct a robotic learning system, it is useful to investigate elements of human learning abilities. No enlightening work exists that deals with the learning theory of human grasping. When discussing this problem, of course, the human hand with its five fingers is considered, which is much more complex than a parallel-jaw gripper as used in this setup. Therefore, in this paper an approach is suggested that is based on our *supposed* human learning abilities when grasping an object. Although no well studied human learning abilities are taken to construct a robotic learning system, the system is used to show that the proposed learning abilities in the field of grasping could in fact be as supposed for a human.

3.1 Local and Global Grasp Criteria

Our work is based on our observation that when humans intend to grasp an unfamiliar object, they mainly consider two criteria on how to choose optimal grasp points. These two criteria are further referred to as *local grasp criteria* and *global grasp criteria*. These two criteria form the basis for the underlying learning system design.

Local Grasp Criteria: A local grasp criterion is mostly independent of a special shape and therefore of global aspects like the distribution of mass of an object. Therefore, it can be applied in the same way to any kind of object. Local criteria are considered first when one decides to grasp an unfamiliar object. Such a criterion is for example to choose a grasp point at two opposite parallel edges.

Global Grasp Criteria: Global grasp criteria, by contrast to the local ones, are strongly interconnected with a special object and therefore seldom to be applied to different kinds. They are considered after the local criteria to find the optimal grasp point. These criteria consider aspects like the distribution of mass of an object, e.g. grasping an object near its center of gravity.

The terms *local* and *global* need some more specifications. The local criteria refer to *local* environmental features near the grasp point whereas the global criteria describe the *global* properties of the position of a grasp point within an object. Therefore, it can be shown that the local criteria are universally valid and the global ones are mostly restricted to a special object.
Technically speaking, the local criteria define an axis on which the grasp point can be searched to further meet the global criteria. For example, the grasp configuration computed in Fig. 1(right) could be determined from the search direction proposed by grasp point 3 in Fig. 1(left). In the learning process these criteria are repeatedly considered one after the other for a finite number of steps until a good grasp point is found. The number of steps varies with the skill of the learner and the habit of the object. For a familiar type of object, the global and local criteria are considered in only one step.
In fact, since the same local criteria can be applied to any kind of object, as mentioned above, they are fully learned prior to the global criteria which will be learned to grasp unfamiliar objects.

3.2 Optimality

A grasp point is optimal according to the local criteria if:

- the fingers can cover the object at this grasp point, and
- no friction occurs between the fingers and the object.

It is considered to be optimal according to the global criteria if:

- no torque occurs between the fingers grasping the object, and
- the object does not slip out of the fingers, and
- the grasp is stable, i.e. the object does not slip between the fingers.

Some sample grasp configurations are shown in Fig. 1.

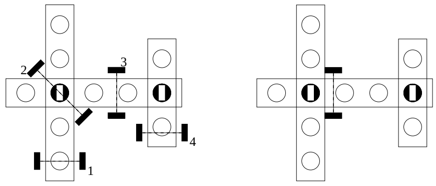

Figure 1: Local and global grasp criteria. Left: some example grasp configurations which are optimal according to the local grasp criteria. Right: the grasp point is optimal according to both criteria.

3.3 Higher Level Criteria

An additional and *higher level criterion* for human grasps is the role of the grip, i.e. the role it plays in order to do further operations, e.g. grasping a cup at its bail in order to drink something or a sledge at its handle to bang a nail into the wall[1]. Other higher level criteria are for example the material or surface of an object. To consider these criteria additional sensors or sophisticated image processing techniques ought to be integrated. However, this is beyond the scope of this work. Our objective is to emulate the abilities of an infant who just intends to get hold of an object as good as possible.

4 Two-Learner System

The criteria mentioned above suggest a system consisting of two learners, one for the local and the other for the global grasp criteria. The states for the first learner only provide the local features $s = (f_{l_1}, \ldots, f_{l_m})$. The learner tries to map them to actions consisting of a rotational component $a = \phi$. The second learner tries to map states of global features $s = (f_{g_1}, \ldots, f_{g_n})$ to actions of translational components: $a = (x, y)$. Because the local criteria are mainly covered from the relative orientation of the gripper, the responsible learner is called *orientation learner*. The global criteria are determined by the position of the grasp point in the object and therefore the proper learner is further referred to as *position learner*. These two learners operate right after each other (Algorithm 1), as a human being is supposed to. The local and global features used in our system are shown in Fig. 2. The

Algorithm 1 Algorithm for learning an optimal grasp point

choose an initial grasp point configuration
$steps \leftarrow 0$
repeat
  $steps \leftarrow steps + 1$
  repeat
    learn with the orientation learner
  until [the grasp point is optimal according to orientation OR number of episodes exceeds a given value]
  repeat
    learn with the position learner
  until [the grasp point is optimal according to the position in the object OR number of episodes exceeds a given value]
until [the optimal grasp point is found OR $steps > steps_{max}$]

[1]For this higher level criteria, aspects of optimality like reducing torque must possibly be shelved.

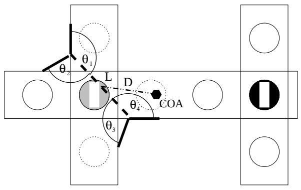

Figure 2: State coding for the learners. The orientation learner uses length L and angles $\Theta_1, \ldots, \Theta_4$ while the position learner integrates the distance D between the center of the grasp-line and the center of area of the object's image.

first component of the state vector of the orientation learner is the length L of the grasp-line. With this feature, good grasp points are distinguished from those which are not adequate because the gripper cannot cover the object. The remaining features are the corresponding angles $\Theta_1, \ldots, \Theta_4$ between the grasp-line and the flanking straight line segments gained by a simple contour tracking process. The features for the position learner are the distance D between the center of the grasp-line and the center of area of the object's image and the torque T around the normal vector $\vec{n}$ of the gripper. Due to the learner separation the local criteria need not be learned for every new object. The orientation learner is a universal learner, which means that the same learner can be used for every object. So this learner will for example learn to grasp objects at opposite parallel or concave edges.
In principle, a two-learner system design is not a new approach [5]. The new aspect of this work is the intention for the use of these two learners. As described above this design was chosen in respect to the local and global criteria and their generalization properties.

5 Self-Valuation

The presented system is self-valuing, a method to gain estimation for the learning algorithms. Self-valuation is done via a force/torque sensor and several image processing techniques. It is important to mention that no optimal grasp point is pre-known. The system finds its own grasp points taking into account the optimality conditions.

5.1 Orientation Learner

The best estimation of a good grasp, determined by the orientation learner, is obtained by the second optimality condition, i.e. no friction at the fingers of the parallel-jaw gripper. When an object slips between or out of the fingers at the moment of closing the gripper, the selected grasp configuration was not optimal according to the local criteria. Some existing systems (e.g. [2]) try to determine the friction occurring within the gripper analytically, i.e. by computing the friction cone via geometrical features. Here, several grasp configurations are tried out with the real robot that values the success or failure of the performed grasp thereafter - like humans who do not analytically compute their optimal grips, but learn by success and failure. Because a parallel-jaw gripper, as used in this work, is very rigid and does not slip like human fingers at the object's surface, friction appears either as a rotation or as a displacement of the object itself. So the valuation signal for the orientation learner is basically observed by image processing. A penalty for self valuation is computed as follows:

$$P = \begin{cases} -(\Theta_{\text{diff}} + \mathcal{D}_{\text{diff}}) & \text{if grip was successful} \\ -\mathcal{P}_{\text{const}} & \text{otherwise} \end{cases}$$

where Θ_{diff} is the angle between the initial and the least inertia axis after the performed grasp, D_{diff} the displacement of the center of area and P_{const} a high constant penalty. Fig. 3 shows a grasp configuration which results in a rotation of the object itself. If a grasp has totally failed and so the first optimality condition cannot be met[2], a predefined penalty is given. The decision if something is in the gripper after a performed grasp is made with the help of the force sensor signals. Fig. 4 shows that the force in direction of the approach vector rises suddenly while lifting up the object.

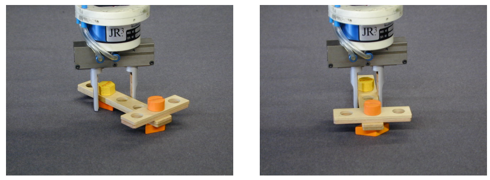

Figure 3: Friction of the fingers result in rotation of the object.

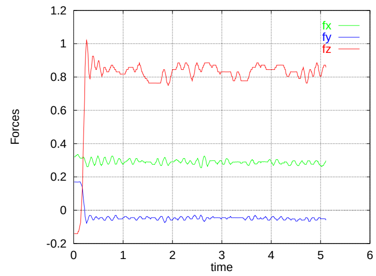

Figure 4: Force profiles during the lift-up process of an object as in Fig. 5(right).

5.2 Position Learner

While the valuation technique for the orientation learner is primarily based on processing images from the camera sensors, the self-valuation of the position learner is primarily gained via the force/torque sensor. The three points for optimality of the global grasp criteria, mentioned above, are taken into account for self-valuation in the following fashion:

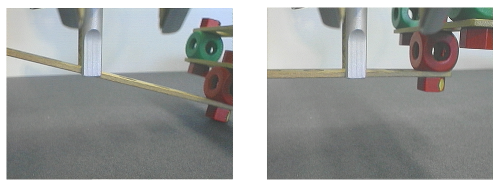

Figure 5: Grips that are suboptimal according to the optimality conditions.

[2] This occurs either when the orientation of the gripper does not permit to cover the object or the object slips out of the fingers while closing them.

Stable Grasp: The grip is stable, according to the optimality conditions, if the grasped object does not move between the fingers of the gripper. This occurs especially when a heavy object is grasped far away from its center of gravity. The gripper is perhaps not strong enough to fix the object at this position. Such a situation is shown in Fig. 5(left). This lift-up movement of the manipulator results in forces shown in Fig. 6. Nearly during the whole lift-up movement, the force in the direction of the approach vector $\vec{a}$ is approximately constant. In the moment when the object loses contact with the table (in this example at $4s$) the force rises to a higher value. This attitude can be evaluated and used within the learner, e.g. this situation is valued with a predefined high penalty to express that such grips are not desired.

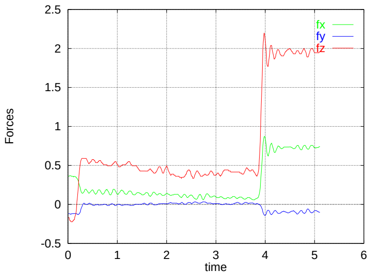

Figure 6: Force profiles when the grip is not stable as shown in Fig. 5(left).

Slipping: Slipping of an object out of the fingers of the gripper is undesirable. This effect mostly occurs as a consequence of an unstable grasp as described above. In such a situation the force in direction of $\vec{a}$ suddenly reduces to zero and the grasp can be thought to have failed. Such a grasp is totally undesirable. Therefore, a constant penalty is given to prevent the system from taking this grip in the future.

Reducing torque: The goal of the position learner is to reduce torque within the fingers of the gripper. Fig. 5(right) shows an example of a grip that produces a large torque. The torque profiles are shown in Fig. 7. Immediately after the beginning of the lift-up process, the torque around the normal vector $\vec{n}$ rises to a value precisely different from zero and stays constant while the object is being held. This torque is computed, negated and directly used with the position learner. Here, no constant penalty is given because a grip with large torque is not necessarily bad. The system must have the possibility to distinguish between grasp points with different torques and choose the best among them.

6 Generalization

The orientation learner is fully applicable to any kind of object, i.e. it provides a total generalization potential. In other work, where the learning process is not divided into two separate learners, the generalization is only partial. This results

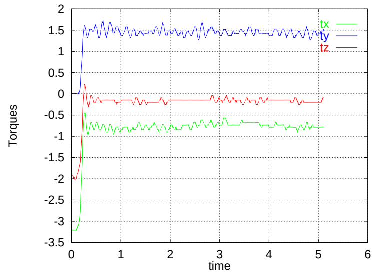

Figure 7: Torque profiles of the grip in Fig. 5(right)

in slower learning phases for new objects. Propositions like "grasping at parallel edges is always good" cannot be made by such systems at all. Here, once the orientation learner has learned several grasping situations, it can be used with any kind of new object the robot is faced with.
The case of the position learner is more complicated. Because of different shapes of objects the global positions of the learned grasp points cannot be applied to every object. Three main subproblems must be solved:

1. In which situation can a previously learned position learner be fully adopted to a new object?

2. When can a previously learned position learner be used as a basis for a new object?

3. When must a completely new position learner be initiated?

The idea to be studied is the object hierarchy. This is a challenging task, but can improve generalization functionality among different kinds of objects. What we are examining is, if one has found suitable features for describing sub-objects, a complex aggregate can be coded in a tree consisting of these sub-objects. Then, a tree distance model, as for example proposed in [7] and [8], can be used to compare several objects. In this manner one can determine the "most similar" object out of the set of previously learned aggregates for a new object. The three subproblems mentioned above can then be solved for example as follows:
Let $\mathcal{L}_s = \{(o_i, l_i) | i = 1 \ldots n\}$ be the set of tuples of n previously stored objects o_i in tree notation, together with their stored position learners l_i, and $dist(o_i, o_k)$ the distance of the trees according to a distance measure. Then,

1. a previously learned position learner l' of an object o' can be fully adopted to a new object o, if $\forall (o_i, l_i) \in \mathcal{L}_s \backslash (o', l')$:
$$dist(o', o) \leq dist(o_i, o) \leq D_{min};$$

2. a previously learned position learner l' of an object o' can be used as basis for a new object o , if $\forall (o_i, l_i) \in \mathcal{L}_s \backslash (o', l')$:
$$D_{max} \geq dist(o', o) \leq dist(o_i, o) > D_{min};$$

3. a completely new position learner is initiated for a new object o, if $\forall (o_i, l_i) \in \mathcal{L}_s$
$$dist(o_i, o) > D_{max},$$

where D_{max} and D_{min} are adequate thresholds for accepting and refusing an object to be equal, respectively.

7 System Overview

7.1 Hardware Configuration

The physical set-up of this system consists of the following components:

Main actuator: One 6 d.o.f. PUMA-260 manipulator is installed overhead in a stationary assembly cell. On the wrist of the manipulator, a pneumatic jaw-gripper with integrated force/torque sensor and "self-viewing" hand-eye system (local sensors) is mounted. The robot is controlled by RCCL (*Robot Control C Library*).

Objects: Most kind of objects are constructed from *Baufix* elements, wooden toys for children containing parts like screws, ledges and cubes. Therefore, these objects are also referred to as *aggregates*. An advantage of these parts is that one can construct very quickly several aggregates that can be tested with the system.

The learning was implemented using a CMAC function approximator [9]. [10] showed that it is a very good and robust technique for dealing with continuous state and action spaces.

7.2 Accumulating Trails

A practical problem that arises is that the system will learn a path from an initial state, i.e initial grasping configuration, up to a final state, i.e. a successful grip. To overcome this side effect in systems where the goal state and not the trajectory leading to it is the most important outcome, we propose an easy new approach for increasing performance of such a learning system, called *accumulating trails*. When a learning system learns a type of path from an initial state to a final state, i.e. by applying a set of actions $a_0, \dots a_n$ to s and its successors, it is sometimes possible to get to the same goal state if applying a set of actions $a'_0 \dots a'_m$ to the state s and its successors, where $m < n$. That is to say, that one would reach the goal state $n - m$ steps earlier.

Let ψ denote the function applying an action a to a state s, denoted $\psi : \mathcal{A} \to (\mathcal{S} \to \mathcal{S})$, where $\mathcal{A}, \mathcal{S}$ are the total sets of actions and states, respectively. The outcome of this function, applying it to an action, is a function on the state space $\mathcal{S}$ called *action execution function*.

Using the definition above, each learning episode can be considered as a composition of functions ($A : \mathcal{S} \to \mathcal{S}$)

$$A(s) = \psi(a_n) \circ \psi(a_{n-1}) \circ \cdots \circ \psi(a_0)(s)$$

where s is the starting state of the episode and a_i is the action applied in time step i. This function composition is further referred to as *sequence*.

A sequence B of action executions $\psi(b_m) \circ \cdots \circ \psi(b_0)$ is called a *sub-sequence* of sequence $A = \psi(a_n) \circ \cdots \circ \psi(a_0)$, if $A(s) = B(s)$:

$$\psi(a_n) \circ \cdots \circ \psi(a_0)(s) = \psi(b_m) \circ \cdots \circ \psi(b_0)(s),\ m \leq n$$

where s is the starting state. Then, sequence A is called *substitutable* through B. The sub-sequence B always produces the same resulting state as the sequence A. That means, if starting in state s it makes no difference whether to "follow" sequence B or sequence A. The state at the end of the sequence is always the same. If a sequence A is not substitutable through any other sequence B, it is called *final*. When the agent's intention is to reach the goal states as soon as possible, as for example in this work[3], the learning algorithm should converge to a situation of only final sequences. An accumulation function on action executions is defined as

$$\oplus : (\mathcal{S} \to \mathcal{S}) \times (\mathcal{S} \to \mathcal{S}) \to (\mathcal{S} \to \mathcal{S}).$$

A sequence $A = \psi(a_n) \circ \cdots \circ \psi(a_0)$ of actions executions is *accumulable*, if

$$\psi(a_n) \oplus \psi(a_{n-1}) \oplus \cdots \oplus \psi(a_0) = B,$$

where B is subsequence of A. The accumulation function describes how to combine action executions to produce shorter sequences. This function has to be defined according to the learning system one wants to develop. The accumulation is defined on action executions and not solely on actions, because it depends on the states if such an accumulation can be performed. In some situations the accumulation function is defined as follows:

$$\psi(a_k) \oplus \psi(a_l) = \psi(a_k \diamond a_l), \tag{1}$$

where $\diamond$ is a function $\diamond : \mathcal{A} \times \mathcal{A} \to \mathcal{A}$.

In most situations, the accumulation function must include a kind of model of the environment and this is only possible by also taking into account the states rather than only the actions as supposed by Eqn. (1). The agent must "know" in which situations it is possible to accumulate action executions and in which situation it is not. However, for some tasks Eqn. (1) is an easy and sufficient definition.

As an example, for application within the orientation learner the accumulation function $\diamond$ is defined as:

$$a_k \diamond a_l = \begin{cases} a_k + a_l & \text{if} \quad -90 \leq a_k + a_l \leq 90 \\ a_k + a_l + 180 & \text{if} \quad a_k + a_l < -90 \\ a_k + a_l - 180 & \text{if} \quad a_k + a_l > 90 \end{cases}$$

assuming that the actions of the orientation learner are rotational movements from the interval $[-90, \dots, 90]$.

8 Experimental Results

To get a uniform and matchable view of the objects, the system learns to grasp, the manipulator initially moves itself over the object so that the x-axis of the cameras coordinate system appears parallel to the axis of least inertia of the object and

[3] It is desirable to find an optimal grasp point as soon as possible.

the center of area in the right side of the image. The center of the object's bounding box coincides with the center of the image. An additional tool-transformation is performed, so that the camera is moved in direction to the working surface.
Several objects were used to test the performance of the whole system. Some of them are shown in Fig. 8. The robot has

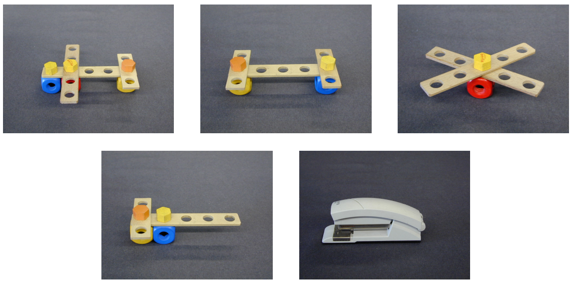

Figure 8: Sample objects.

found a good and stable grasp point for each object that fulfills the optimality conditions given above, most times near the object's center of gravity. Two special results of a grasping operation are shown in Fig. 9. In Fig. 9(left) the manipulator grasped the object at a point different from the center of area but near the center of mass of the object. Fig. 9(right) shows a successful grasp at a convex edge of a different object. To

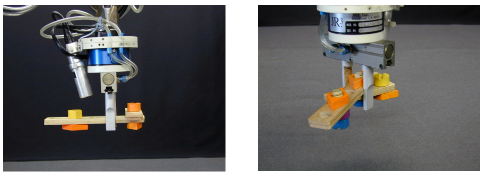

Figure 9: Successfully performed grasping operations.

show the generalization ability of the orientation learner, it was first applied to a new object until a defined number of epoches. Thereafter, the same learner was used on a different object to show that the average steps until the goal state decrease much faster. The result is shown in Fig. 10. In the second part of the experiment the orientation learner did not start at the average steps of 3 where the initially performed orientation learner ended. This is due to the fact that in the first cycle a simple ledge was used and the learner still not converged while in the second cycle a more complex object was used. However, one can see that in the second cycle the orientation learner was quicker. Only new states that do not occur on the simple ledge have to be learned additionally.

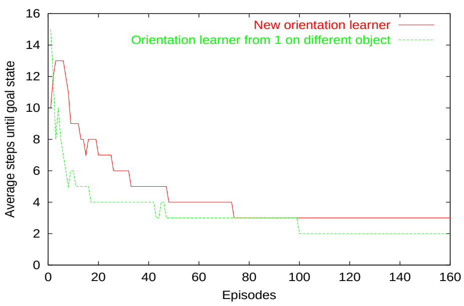

Figure 10: Generalization of the orientation learner.

9 Discussion and Future Work

We presented a self-valuing learning system that is capable of grasping various kind of objects. Our system consists of two learners based on local and global grasping criteria. These criteria are supposed to imitate human learning abilities in the field of grasping. While the orientation learner is applicable to arbitrary objects and therefore fully generalizes between them, the position learner is mostly dependent on a special object and its physical properties. The system shows the ability to grasp several kind of objects and to generalize the learned faculties to new ones. An interesting future work is to adopt the presented system to a multi-fingered robot hand. With such a hand a single grasp point is much more complex than with a parallel-jaw gripper. Furthermore, the possible actions of the learners are more challenging. However, the basic principle of two learners, based on local and global criteria, and the self-valuing approach could be maintained.

References

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

[2] G. Smith, E. Lee, K. Goldberg, K. Boehringer, and J. Craig, "Computing parallel-jaw grips," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 1897–1903.

[3] P. J. Sanz A. Morales, G. Recatalá and Ángel P. del Pobil, "Heuristic vision-based computation of planar antipodal grasps on unknown objects," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2001.

[4] I. Kamon, T. Flash, and S. Edelman, "Learning to grasp using visual information," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996, pp. 2470–2476.

[5] J. Zhang, G. Brinkschröder, and A. Knoll, "Visuelles Reinforcement-Lernen zur Feinpositionierung eines Roboterarms über kompakte Zustandskodierung," in *Tagungsband Autonome Mobile Robotersysteme*, München, 1999.

[6] M.-C. Nguyen and V. Graefe, "Self-learning vision-guided robots for searching and grasping objects," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 1633–1638.

[7] K. C. Tai, "The tree-to-tree correction problem," *J. Assoc. Comput. Mach.*, vol. 26, no. 3, pp. 422–433, 1979.

[8] L. Wang T. Jiang and K. Zhang, "Alignment of Trees - An Alternative to Tree Edit," *Theoretical Computer Science*, vol. 143, no. 1, pp. 137–148, 1995.

[9] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller.," *Journal of Dynamic Systems, Measurement, and Control, 97, 220–227.*, 1975.

[10] J. Santamar, R. Sutton, and A. Ram, "Experiments with reinforcement learning in problems with continuous state and action spaces," *Adaptive Behavior 6(2):163-217, 1998.*, 1998.