2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning

Staffan Ekvall and Danica Kragic*
Computational Vision and Active Perception Laboratory
Centre for Autonomous Systems
Royal Institute of Technology, Stockholm, Sweden
ekvall, danik@nada.kth.se

Abstract— **In this paper, we address the problem of automatic grasp generation for robotic hands where experience and shape primitives are used in synergy so to provide a basis not only for grasp generation but also for a grasp evaluation process when the exact pose of the object is not available. One of the main challenges in automatic grasping is the choice of the object approach vector, which is dependent both on the object shape and pose as well as the grasp type. Using the proposed method, the approach vector is chosen not only based on the sensory input but also on experience that some approach vectors will provide useful tactile information that finally results in stable grasps. A methodology for developing and evaluating grasp controllers is presented where the focus lies on obtaining stable grasps under imperfect vision. The method is used in a teleoperation or a Programming by Demonstration setting where a human demonstrates to a robot how to grasp an object. The system first recognizes the object and grasp type which can then be used by the robot to perform the same action using a mapped version of the human grasping posture.**

I. Introduction

In the field of Programming by Demonstration (PbD), [1], [2], the user teaches the robot new tasks by simply demonstrating them. The robot can initially imitate human behavior and then improve through continuous interaction with the environment. For task learning by instruction, complex systems that involve object grasping and manipulation, visual and haptic feedback may be necessary. The robot has to be instructed *what* and *how* to manipulate, [3]. If the kinematics of robot arm/hand system is the same as for the human, a one-to-one mapping approach may be considered. This is, however, seldom the case. The problems arising are not only related to the mapping between different kinematic chains for the arm/hand systems but also to the quality of the object pose estimation provided by the vision system.

Considering specifically object manipulation tasks, the work on automatic grasp synthesis and planning is of significant relevance, [4], [5], [6], [7]. The main issue here is the automatic generation of stable grasps assuming that the model of the robot hand is available and that certain assumptions about the shape of the object can be made. Example of assumptions may be that the full and exact pose of the object is known in combination with its (approximate) shape, [4]. Another common assumption is that the outer contour of the object can be extracted and a planar grasp applied, [6]. The work on contact-level grasps synthesis concentrates mainly on finding a fixed number of contact locations with no regard to hand geometry, [8], [9]. Taking into account both the hand kinematics as well as some *a-priori* knowledge about the feasible grasps has been acknowledged as a more flexible and natural approach towards automatic grasp planning. The method proposed in [4] presents a system for automatic grasp planning for a Barrett hand by modeling an object as a set of shape primitives, such as spheres, cylinders, cones and boxes in a combination with a set of rules to generate a set of grasp starting positions and pregrasp shapes. Similar to our work, [10], [11] also base the choice of grasp on a human demonstration. However, the problem considered is somewhat different. In our work, we do not assume that the objects are placed at the exact location as during the demonstration. Therefore, only the grasp type information can be used, not the approach direction of the human hand. Hence, the robot has to find a suitable approach vector either by calculation or by performing an exhaustive search. The latter alternative is used in this paper.

The contribution of the work presented in this paper arises from the following problems stated above:
i) A grasp is related to object pose as well as its shape and not only to a set of points generated along its outer contour. This means that we do not assume that the initial hand position is such that only simple grasps (e.g. planar grasps) can be executed as proposed in [6]. In addition, grasps relying only on a set of contact points may be impossible to generate online since the available sensory feedback may not be able to detect the same points once the pose of the object has changed.
ii) We use *experience* provided by the human teacher to model a set of most likely hand preshapes with respect to the object. Similar idea was investigated in [4] but only one robotic hand and four grasp preshapes were considered. We evaluate both Barrett [12] and Robonaut [13] hands and grasp preshapes are generated based on recognition of human grasps which makes them more natural. This is, of course, of interest for humanoid robots where the current trend is to resemble human behavior as close as possible.
iii) Finally, we also evaluate the quality of different grasp types with respect to inaccuracies in pose estimation. This is

This work has been supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657.

an important issue that commonly occurs in robotic systems. The reasons may be that the calibration of the vision system or hand-eye system is not exact or that a detailed model of the object is not available. We evaluate how large position estimation error different grasp types can handle.

This paper is organized as follows. In Section II, the required system components are briefly described. Section III presents our grasp mapping strategy, and Section IV and V covers the grasp controllers and grasp planning. In Section VI the approach is thoroughly tested, and Section VII concludes the paper.

II. System Description

The development and evaluation of grasping sequences is considered as a part of a PbD framework. The human and the robot are both standing in front of a table, on which a set of objects are placed. The human demonstrates a task to the robot by moving objects on the table. The robot recognizes which objects have been moved and where using visual feedback. Using magnetic trackers placed on the human hand, the robot is also able to recognize which grasp type has been used to move a specific object. Based on this, the robot is able to reproduce the task performed by the human, [14]. Objects may, but are not required to be placed at the same location as during the demonstration. Our more recent work has also evaluated how tasks can be learned based on multiple demonstrations, [15].

In this study, we design and evaluate a system for automatic grasp generation and planning that can be used in the above scenario to facilitate grasping of new objects that are in shape similar to the objects that the robot has already learned how to grasp. This closely relates to the use of shape primitives, [4]. Here, we evaluate our approach given two kinematically different hands, the Barrett hand and the Robonaut hand.

To execute a task in a PbD framework, the robot first has to be able to interpret the actions performed by the human operator, then map them to its own frame of reference and repeat the task. We shortly review the components currently used in our system:

1: Object Recognition and Pose Estimation - By estimating the objects' poses before and after an action, the system can identify *which* object has been moved and *where*. For object recognition and pose estimation, we use Receptive Field Cooccurrence Histograms(RFCH) [16], [17]. In this study, it is assumed that the objects are placed directly on the table and the pose is then represented by three parameters (x, y and ϕ).

2: Grasp Recognition - A data-glove equipped with a set of magnetic trackers provides hand postures to to the grasp recognition system as described in [18]. Also, the position of the hand is used to segment each movement action. To make the sequencing and task recognition easier, an object manipulation action is considered complete if the hand of the demonstrator moves out of the working area.

3: Grasp Mapping - Depending on the target robot, i.e. which type of robot hand is used, a fixed defined grasp mapping scheme maps the human grasps to robot grasps as presented in Section III.

4: Grasp Execution - The robot selects a grasp controller depending on the desired grasp type. The controller will choose the best available approach direction the maximizes the probability of successful grasp. This choice is based on a grasp planning method, described in Section V.

The evaluation of the system proposed in this work is performed using GraspIt! [19], to allow for repetitive experiments and statistical evaluation. The obtained results will facilitate further development of the robot grasping system presented previously in [20].

III. Grasp Mapping

Related to grasp planning at large, it has been argued that one procedure that can limit the large number of possible robot hand configurations is to use grasp preshapes. This is strongly motivated by the fact that even humans when planning a grasp, unconsciously simplify this by selecting one of only a few different prehensile postures. These prehensile postures are those appropriate for the object and for the task to be performed, [21]. Before grasping the object, the human grasp type first has to be mapped to a similar robot grasp type. For this purpose, a mapping scheme showed in Fig. 1 was defined. It has to be noted here that the names of the robot grasp types do not refer only to hand postures, but instead to grasping controllers. A grasping controller takes into consideration both the final posture of the hand as well as the object approach strategy used by the human. Hence, different strategies are used to grasp an object dependent on the grasp type.

IV. Grasp Controllers

There are two basic grasp controllers in the system: Power Grasp and Precision Grasp. There are eight variations of these, three for the Barrett hand and five for the Robonaut hand. Among these variations, it is only the initial hand posture and the finger closing procedure that varies, the underlying controller is the same. The two basic controllers are described below.

- **Power Grasp** - First, the initial hand posture is set according to the grasp type recognized from the human demonstrator. Then the hand approaches the object with the palm towards the object. Once a contact is detected, all fingers close simultaneously. Dependent on the grasp type, the joint angle speed may be different for each joint, causing for example the thumb to close more slowly. When a fingertip contact is detected, that finger stops its closure, and once all fingers have stopped, force is applied and the object is grasped. This controller type will in general give a contact at the palm that results in a more stable grasp.
- **Precision Grasp** - This controller is similar to the Power Grasp, but with an added dimension. Once a contact is detected, typically at one of the fingertips, the hand can retract a predefined distance and then close all

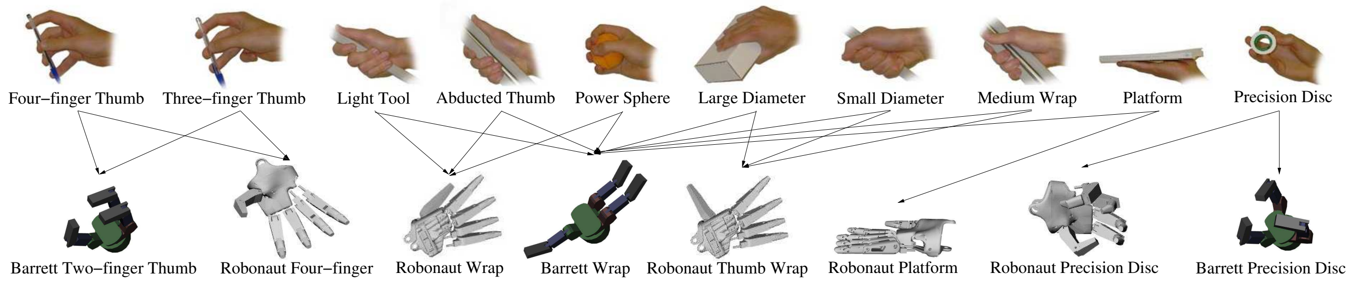

Fig. 1. Initial robot hand postures for different grasp types.

fingers simultaneously. This allows the robot to better combine tactile sensing with computer vision, as we previously demonstrated in [20].

The grasp approach vector describes the 3D angle that the robot hand approaches the object with, relative to the object's pose. For both controllers above, a contact displacement controller, such as the one described in [7], would increase the grasp success rate. After the robot has made an initial contact with the object, the fingers are displaced by a small amount in the direction of the negative error gradient. However, this can be considered as a separate problem and is outside the scope of this study. Another way of improving performance is to add a more advanced controller, such as the one presented in [22]. Here, the estimated pose combined with tactile sensing is used as feedback to adjust the forces applied at the finger joints.

V. Grasp Planning

In the current system, the planning is performed for different objects and two robot hands in the grasping simulator GraspIt! [19]. An object, considered as a shape primitive with known pose, is approached with the robot hand from a set of initial positions. Then, a grasp quality measure is used for estimating the quality of the grasp obtained from every initial position. We use the quality measures provided by GraspIt!, see [19] for details. All results are stored in a so called *grasp experience* database. Our approach to grasp planning is related to the work described in [19] and [23]. However, those approaches require the object to be composed of object primitives, explained later in Section V-B. For each primitive, a set of predefined candidate grasps and approach directions are evaluated. In this work, we do not use predefined approach vectors but instead evaluate many approach vectors for each object. The vectors may target the center of the object or one of the primitives, but we believe the information gained from previous grasping of a primitive has limited value when the primitive is attached to an object. The most significant contribution compared to the above papers is that we evaluate how the grasps perform under imperfect pose estimation. We also evaluate how the primitive representation affects the results once the real object is grasped.

In our approach, similar to [4], we use the grasp controllers described earlier in order to resemble the real world conditions as good as possible. Also, we plan not only for each object and end-effector, but also for each grasp type. Naturally, the best approach vector for a power grasp is not necessarily the best approach vector for a precision grasp.

For power grasps, three parameters (θ, ϕ, ψ) are varied describing the approach direction and hand rotation. For precision grasps, a fourth parameter d, that describes the retract distance when contact is detected, is added. The number of evaluated values for the variables are θ=9, ϕ=17, ψ=9, d=6. For the precision grasps the search space was hence 8262 grasps which required about an hour of training.

A. Grasp Retrieval

At the run-time, the robot retrieves the approach vector that has the highest probability of success, from the grasp experience database. Also, if it is given the option to choose from several different grasp types, the grasp quality will in general be even higher. However, because of robot kinematic constraints and possible non-free paths toward the object, all approach directions are not suitable at task execution time. Thus, the robot searches the database only for directions that are applicable to the situation at hand.

B. Training on Object Primitives

As mentioned earlier, a model of each object is necessary for training the grasp planner. It is not likely that the robot will be able to acquire such models automatically especially if the shape is very complex. However, it is realistic to assume that it will be possible to extract *shape primitives* using computer vision or laser technology. The idea is to represent each object by its appearance (textural properties) and shape primitives. The appearance is used for recognition, and the primitives for grasp planning. Several primitives build up an object and the primitives can be of different basic shapes: truncated cone, sphere, box, cylinder etc. Recent progress presented in [24] show a promising method for retrieving shape primitives using vision, although the method currently is restricted to objects with uniform color. To evaluate an object representation using primitives, we have designed a primitive representation for each object in our scenario, see Fig. 2. For evaluation purposes, we have modeled objects as presented in Fig. 2.

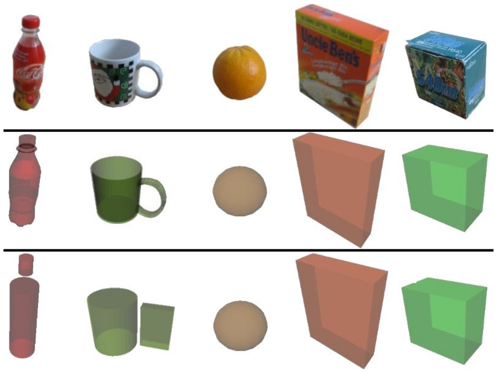

Fig. 2. First row: The real objects. Second row: The modeled objects. Third row: The object primitives used for training.

VI. Experimental Evaluation

In this section, we provide both i) qualitative experiments that show grasps performed by the robot hand given the current state of the environment and *grasp experience* database, and ii) quantitative experiments that show how small errors in pose estimation affect the success of the final grasping result.

Five objects shown in Fig. 2 were modeled and added to the GraspIt! simulator. The real objects were placed on a table, Fig. 3 (left). A camera mounted on a side of the table monitors the world state which consist of five objects placed at arbitrary positions. In this example, the objects are placed at the same positions in the robot world as in the human world. However, as the robot calculates the approach vector and performs pose estimation, it can also perform the task when the objects are placed in a different configuration compared to the human world. This setting, however, causes problems when object collisions may occur in one world and not in the other. In [15] we solve this by invoking *planning* into the PbD process.

The human teacher, wearing a data-glove with magnetic trackers, starts by moving one of the objects. The object moved is recognized by the vision system as well as the grasp human used to move it. This information is then used to first generate a suitable robot grasp that controls the movement of the robot hand in the simulator. References to our previous work that describe this in detail are provided in Section II.

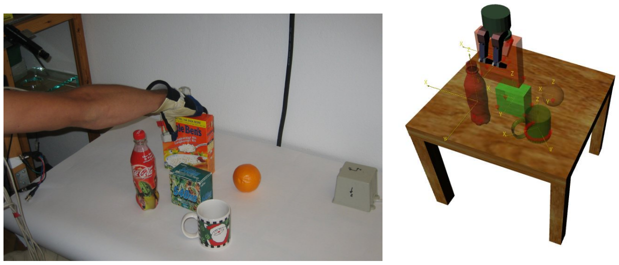

Fig. 3. Left: The human moves the rice box. The system recognizes what object has been moved and which grasp is used. Right: The robot grasps the same object using the mapped version of the recognized grasp.

Fig. 4 shows a few examples of the best grasps obtained when the robot is free to choose any approach direction. Fig. 4(h) shows an example of a failed grasp, due to a simulated error in pose estimation.

The success rate of this system depends on the performance of three modules: i) object recognition, ii) grasp recognition and iii) pose estimation of the grasped object. As demonstrated in previous papers, [16], [18], the object recognition rate for only five objects is around 100 %, and the grasp recognition ratio is about 96 % for ten grasp types. Therefore, the performance in a simulated environment may be considered close to perfect, since the exact pose of each object is considered known. However, this will usually not be the case if either the robot hand-eye system is not perfectly calibrated or if the perfect model of the object is not available. Therefore, we introduce some error in the pose estimation process and once again evaluated the performance of the system.

A. Introducing Error in Pose Estimation

To evaluate the performance under imperfect pose estimation, we have simulated errors by providing an object pose with an offset. As pointed out in [25], the robustness of a grasp to positioning the end-effector has not been widely addressed in the literature.

In our experiment, the target object was placed on the table and the robot performed a grasp 50 times. Each time, a fixed-length vector with a random orientation was added to the position provided to the robot. As a result, the robot interpreted the situation as if the object had been moved. This was repeated for five different vector lengths: 0, 1, 2, 3 and 4 cm. In total, the robot grasped the object 250 times.

Fig. 5 - 6 show the grasp success rates for various grasps and objects, under increasing error in position estimation. Here, a grasp is considered successful if it is a force-closure grasp. As expected, power grasps are more robust to position errors than precision grasps. The precision grasps target details of an object, e.g., the bottle cap or the ear of the mug. Thus, the grasps are much more sensitive to position inaccuracies. However, a tactile grasp adjustment algorithm would increase the grasp success rate and make the precision grasp quality higher.

It is clear that the Barrett hand is more robust than the Robonaut hand, likely due to its long fingers. The exception is the grasping of the mug, to the right in Fig. 5, where the Robonaut Four-finger Thumb grasp is the best.

The bottle and the mug have been trained both using a primitive model and using the real model (see Fig. 2). Training on the primitive model does not decrease the grasp success rate much, especially not for the bottle. However, the primitive model of the mug is, unlike the real mug, not hollow, which causes problems for some of the precision grasps trained on the primitive.

We have also evaluated how an error in rotation estimate affects the result. For each object and grasp type, we tested how much the object could be rotated before the grasp failed. As expected, for symmetric objects like the orange and the bottle this type of error has no effect. However, for the

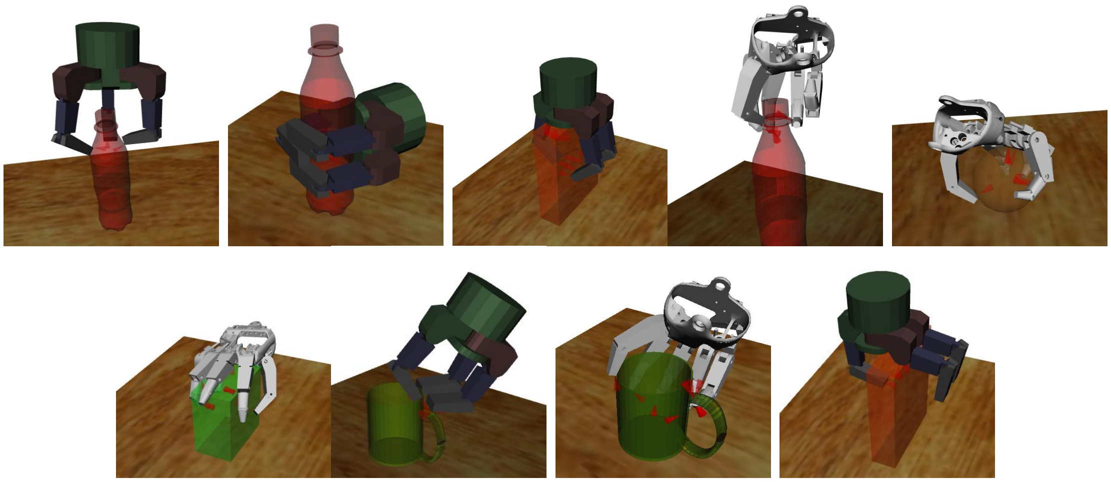

Fig. 4. Examples of grasp executions for various grasp types and objects. From left: (a) Barrett Precision Disc, (b) Barrett Wrap, (c) Barrett Wrap, (d) Robonaut Precision Disc, (e) Robonaut Thumb Wrap, (f) Robonaut Thumb Wrap, (g) Barrett 2-finger Thumb, (h) Robonaut 4-finger Thumb, (i) Barrett Failed Wrap. (a)-(h) shows successful grasps, while (i) shows a failed grasp due to a simulated error in pose estimation. The contact friction cones are plotted in red.

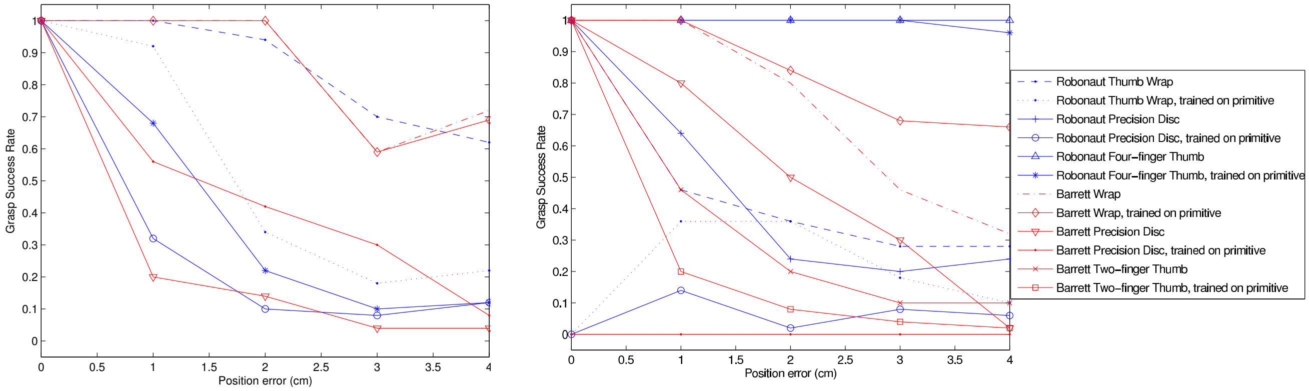

Fig. 5. Left: Grasping the bottle. Right: Grasping the mug.

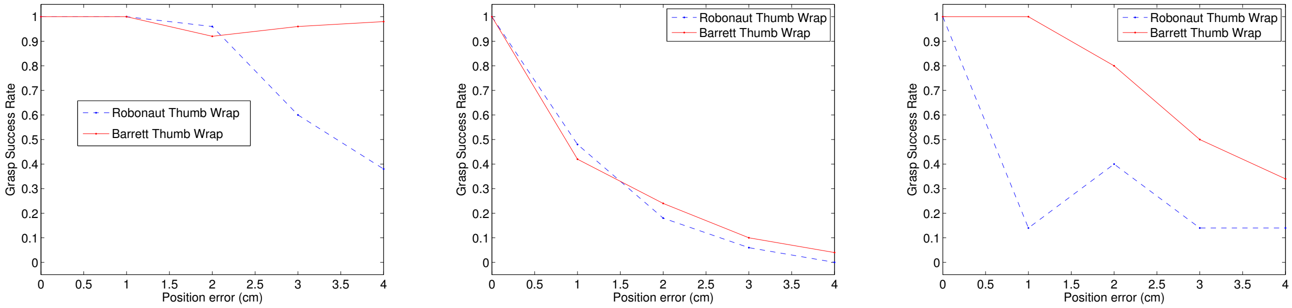

Fig. 6. Left) Grasping the orange, Middle) Grasping the zip disc box, Right) Grasping the rice box.

other objects we found that the difference in rotation error tolerance is large. Table I shows the rotation tolerance for various objects and grasp types. For two of the Robonaut grasps on the mug, the rotation is not a problem, with a perfect success rate. For one of the Barrett grasps on the mug, the rotation estimation is absolutely crucial and cannot withstand a small rotation inaccuracy. Thus, this type of grasp should be avoided for this object.

VII. Conclusions

In this paper, we have presented a method for generating robot grasps based on the shape primitives and human demonstration. One of the main challenges in automatic grasping is the choice of approach vector, which is dependent both on the object and the grasp type. Using the proposed method, the approach vector is chosen not only based on perceptional cues, but on experience that some approach

Object and Grasp Type:	Rot. Err. Tolerance (degrees):
Mug, Robonaut Precision Disc	4
Mug, Robonaut Thumb Wrap	180
Mug, Robonaut Four Finger Thumb	180
Zip Disc Box, Robonaut Thumb Wrap	17
Rice Box, Robonaut Thumb Wrap	2
Zip Disc Box, Barrett Wrap	3
Rice Box, Barrett Wrap	17
Mug, Barrett Wrap	12
Mug, Barrett Precision Disc	0
Mug, Barrett Two Finger Thumb	6

TABLE I
THE ROTATION ERROR TOLERANCE FOR DIFFERENT OBJECTS AND GRASP TYPES.

vectors will provide useful tactile cues that result in stable grasps. Moreover, a methodology for developing and evaluating grasp controllers has been presented. Focus lies on obtaining stable grasps under imperfect vision, something that has not been thoroughly investigated in the literature.

As a part of the future work, it will be interesting to evaluate the proposed strategies in combination with tactile feedback on the robot hand, [20]. The use of the simulation results presented here was necessary for generating insight into the problem as well as performing the statistical evaluation for the grasp experience, since i) the world must be reset after each grasp attempt, and ii) computing the grasp quality measure requires perfect world knowledge. The kinematic simulation results presented in this paper are interesting, but recent experiments indicate that with dynamic simulation, better results are obtained. This is especially true for precision grasps, where it is difficult to obtain a stable grasp without moving the object. Thus, an evaluation using dynamic simulation would probably improve the results. This is part of our future work.

Currently, our method assumes that the robot can detect contact anywhere on the robot hand. A more realistic assumption is that the hand is equipped a set of tactile sensors and contact can only be detected at one of these, [20]. Considering this in the planning and evaluation stage is a part of the current work.

The grasp controllers presented here do not use the contact vector. Recent work presented in [26] show that such sensors are being developed. Thus, more tactile information will be available and should be used in the controller. For example, integration with a contact displacement controller would further increase the grasp success rate.

REFERENCES

[1] H. Friedrich, R. Dillmann, and O. Rogalla, “Interactive robot programming based on human demonstration and advice,” in *Christensen et al (eds.):Sensor Based Intelligent Robots, LNAI1724*, pp. 96–119, 1999.

[2] J. Aleotti, S. Caselli, and M. Reggiani, “Leveraging on a virtual environment for robot programming by demonstration,” in *Robotics and Autonomous Systems, Special issue: Robot Learning from Demonstration*, vol. 47, pp. 153–161, 2004.

[3] S. Calinon, A. Billard, and F. Guenter, “Discriminative and adaptative imitation in uni-manual and bi-manual tasks,” in *Robotics and Autonomous Systems*, vol. 54, 2005.

[4] A. T. Miller, S. Knoop, and H. I. C. P.K. Allen, “Automatic grasp planning using shape primitives,” in *In Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1824–1829, 2003.

[5] N. S. Pollard, “Closure and quality equivalence for efficient synthesis of grasps from examples,” *International Journal of Robotic Research*, vol. 23, no. 6, pp. 595–613, 2004.

[6] A. Morales, E. Chinellato, A. H. Fagg, and A. del Pobil, “Using experience for assessing grasp reliability,” *International Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 671–691, 2004.

[7] R. Platt Jr, A. H. Fagg, and R. A. Grupen, “Extending fingertip grasping to whole body grasping,” in *Proc. of the Intl. Conference on Robotics and Automation*, 2003.

[8] A. Bicchi and V. Kumar, “Robotic grasping and contact: A review,” in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA’00*, pp. 348–353, 2000.

[9] D. Ding, Y.-H. Liu, and S. Wang, “Computing 3-d optimal formclosure grasps,” in *In Proc. of the 2000 IEEE International Conference on Robotics and Automation*, pp. 3573 – 3578, 2000.

[10] S. Kang and K. Ikeuchi, “Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps,” *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 81–95, 1997.

[11] R. M. Voyles and P. K. Khosla, “Gesture-based programming: A preliminary demonstration,” in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 708–713, 1999.

[12] http://www.barrett.com/robot/products/hand/handfram.htm.

[13] C. Lovchik and M. Diftler, “The Robonaut hand: a dexterous robot hand for space,” in *Robotics and Automation, IEEE International Conference on*, vol. 2, pp. 907–912, 1999.

[14] S. Ekvall and D. Kragic, “Integrating object and grasp recognition for dynamic scene interpretation,” in *IEEE International Conference on Advanced Robotics, ICAR’05*, 2005.

[15] S. Ekvall and D. Kragic, “Learning task models from multiple human demonstration,” in *IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN’06*, 2006.

[16] S. Ekvall and D. Kragic, “Receptive field cooccurrence histograms for object detection,” in *IEEE/RSJ IROS*, 2005.

[17] S. Ekvall, D.Kragic, and F. Hoffmann, “Object recognition and pose estimation using color cooccurrence histograms and geometric modeling,” *Image and Vision Computing*, vol. 23, pp. 943–955, 2005.

[18] S. Ekvall and D. Kragic, “Grasp recognition for programming by demonstration,” in *IEEE/RSJ IROS*, 2005.

[19] A. T. Miller and P. Allen, “Graspit!: A versatile simulator for grasping analysis,” in *Proceedings of the of the 2000 ASME International Mechanical Engineering Congress and Exposition*, 2000.

[20] D. Kragic, S. Crinier, D. Brunn, and H. I. Christensen, “Vision and tactile sensing for real world tasks,” *Proceedings IEEE International Conference on Robotics and Automation, ICRA’03*, vol. 1, pp. 1545–1550, September 2003.

[21] J. Napier, “The prehensile movements of the human hand,” in *Journal of Bone and Joint Surgery*, vol. 38B(4), pp. 902–913, 1956.

[22] J. Tegin and J. Wikander, “A framework for grasp simulation and control in domestic environments,” in *IFAC-Symp. on Mechatronic Syst.*, (Heidelberg, Germany), Sept. 2006.

[23] A. Morales, P. Azad, T. Asfour, D. Kraft, S. Knoop, R. Dillmann, A. Kargov, C. Pylatiuk, and S. Schulz, “An anthropomorphic grasping approach for an assistant humanoid robot,” in *International Symposium on Robotics (ISR)*, 2006.

[24] F. Bley, V. Schmirgel, and K. Kraiss, “Mobile manipulation based on generic object knowledge,” in *IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN’06*, 2006.

[25] A. Morales, P. J. Sanz, A. P. del Pobil, and A. H. Fagg, “Vision-based three-finger grasp synthesis constrained by hand geometry,” *Robotics and Autonomous Systems*, vol. 54, no. 6, pp. 494–512, 2006.

[26] G. Cannata and M. Maggiali, “An embedded tactile and force sensor for robotic manipulation and grasping,” in *IEEE-RAS International Conference on Humanoid Robots*, 2005.