Intel Serv Robotics (2009) 2:23–30
DOI 10.1007/s11370-008-0026-3

ORIGINAL RESEARCH PAPER

Demonstration-based learning and control for automatic grasping

Johan Tegin · Staffan Ekvall · Danica Kragic · Jan Wikander · Boyko Iliev

Received: 16 January 2008 / Accepted: 26 May 2008 / Published online: 28 August 2008

Abstract We present a method for automatic grasp generation based on object shape primitives in a Programming by Demonstration framework. The system first recognizes the grasp performed by a demonstrator as well as the object it is applied on and then generates a suitable grasping strategy on the robot. We start by presenting how to model and learn grasps and map them to robot hands. We continue by performing dynamic simulation of the grasp execution with a focus on grasping objects whose pose is not perfectly known.

Keywords Grasping · Learning · Control · Simulation · Robot

1 Introduction

Robust grasping and manipulation of objects is one of the key research areas in the field of robotics. There has been a significant amount of work reported on how to achieve stable and manipulable grasps [2,19,20,24,26]. This paper presents a method for an initial grasp generation and control for robotic hands where human demonstration and object shape primitives are used in synergy. The methodology in this paper tackles different grasping problems, but the main focus is on choosing the object approach vector, which is dependent on both the object shape and pose as well as the grasp type. Using the proposed method, the approach vector is chosen not only based on perceptional cues but also on experience that some approach vectors will provide useful tactile cues that finally result in stable grasps. Moreover, a methodology for developing and evaluating grasp schemes is presented where the focus lies on obtaining stable grasps under imperfect vision.

The presented methodology is considered in a Programming by Demonstration (PbD) framework [1,12], where the user teaches the robot tasks by demonstrating them. The framework borrows ideas from the field of teleoperation that provides a means of direct transmission of dexterity from the human operator. Most of the work in this field focuses however on low-level support such as haptic and graphical feedback and deals with problems such as time delays [17]. For instruction systems that involve object grasping and manipulation, both visual and haptic feedback are necessary. The robot has to be instructed *what* and *how* to manipulate. 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, never 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 delivered by the vision system. Our previous results related to these problems have been presented in for example [9] and [7].

The contributions of the work presented here are as follows:

1. Suitable grasps are based on object pose and 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 planar grasps can be

J. Tegin (✉) · J. Wikander
Mechatronics Laboratory, KTH Machine Design, Brinellvägen 83, 100 44 Stockholm, Sweden
e-mail: johant@md.kth.se

S. Ekvall · D. Kragic
Computational Vision and Active Perception Laboratory, KTH NADA/CVAP, Stockholm, Sweden

B. Iliev
Center for Applied Autonomous Sensors Systems, Örebro University, Örebro, Sweden

executed as proposed in [20]. In addition, grasps relying only on a set of contact points may be impossible to generate on-line, since the available sensory feedback may not be able to estimate the exactly same points on the object's surface once the pose of the object is changed.
2. The choice of the suitable grasp is based on the *experience*, i.e., it is learned from the human by defining the set of most likely hand preshapes with respect to the specific object. A similar idea, also using the BarrettHand [30], was investigated in [19]. Grasp preshapes are generated based on recognition of human grasps. This is of interest for humanoid robots where the current trend is to resemble human behavior as closely as possible.
3. Finally, we evaluate the quality of different grasp types with respect to inaccuracies in pose estimation. This is 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 big pose estimation error different grasp types can handle.

1.1 Related work

The work on automatic grasp synthesis and planning is relevant to the ideas presented here [19,26,20,24]. In automatic grasp synthesis, it is commonly assumed that the position, orientation, and shape of the object is known [19]. Another assumption is that it is possible to extract the outer contour of an object and then apply a planar grasp [20]. The work on contact-level grasp synthesis concentrates mainly on finding a fixed number of contact locations without considering the hand design [2,6].

Taking into account hand kinematics and a priori knowledge of the feasible grasps has been acknowledged as a more flexible and natural approach toward automatic grasp planning [19,25]. In [25], a method for adapting a given prototype grasp of one object to another object, such that the quality of the new grasp would be at least 75% of the quality of the original, was developed. This process, however, required a parallel algorithm running on supercomputer to be computed in reasonable time. This clearly shows the need to reduce the solution space for grasping problems to reach solutions in an acceptable time.

The method proposed in [19] presents a system for automatic grasp planning for a BarrettHand by modeling objects as sets of shape primitives; spheres, cylinders, cones, and boxes. Each primitive is associated with a set of rules to generate a set of candidate pregrasp hand positions and configurations. Human demonstrations can also be used to find a suitable grasp, especially focusing on the grasp approach [27]. Examples of robotic manipulation include [4,21,23].

2 System description

In our work, robotic grasping is performed by combining a PbD framework with semiautonomous grasping and performance evaluation. We assume that a task such as *pick up/move/put down* an object is first demonstrated to the robot. The robot recognizes which object has been moved, as well as where, using visual feedback. The magnetic trackers on the human hand provide information that enables the robot to recognize the human grasp. The robot then reproduces the action [9]. The approach is evaluated using a modified and extended version of the robot grasp simulator GraspIt! [18] to allow for repetitive experiments and statistical evaluation. In the simulation experiments, we use the BarrettHand and a hybrid force/position control framework. It is shown how dynamic simulation can be used for building grasp experience, for the evaluation of grasp performance, and to establish requirements for the robot system.

The current components in our PbD system are as follows:

1. Object recognition and pose estimation: estimating the pose of an object before and after an action enables the system to identify *which* object has been moved *where*. For object recognition and pose estimation, receptive field co-occurrence histograms are used [8,10]. It is assumed that the objects are resting on a table and can be represented by parameters (x, y, and ϕ).
2. Grasp recognition: a data-glove with magnetic trackers provides hand postures for the grasp recognition system [7].
3. Grasp mapping: an off-line learned grasp-mapping procedure maps human to robot grasps as presented in Sect. 2.1.
4. Grasp learning: testing many grasps in simulation allows for a quantitative rating of each grasp. This rating is used to select an appropriate grasp controller. Here, the object will be approached from the direction that maximizes the probability of reaching a successful grasp. This is presented in more detail in Sect. 4.
5. Grasp execution: a semiautonomous grasp controller is used to control the hand from the planned approach position until a force closure grasp is reached (Sect. 3).

2.1 Grasp mapping

It has been argued that grasp preshapes can be used to limit the large number of possible robot hand configurations. This is motivated by the fact that—when planning a grasp—humans unconsciously simplify the grasp choice by choosing from a limited set of prehensile postures appropriate for the object and task at hand [22]. Cutkosky [5] classified human grasps and evaluated how the task and object geometry affect the choice of grasp. The work on virtual fingers

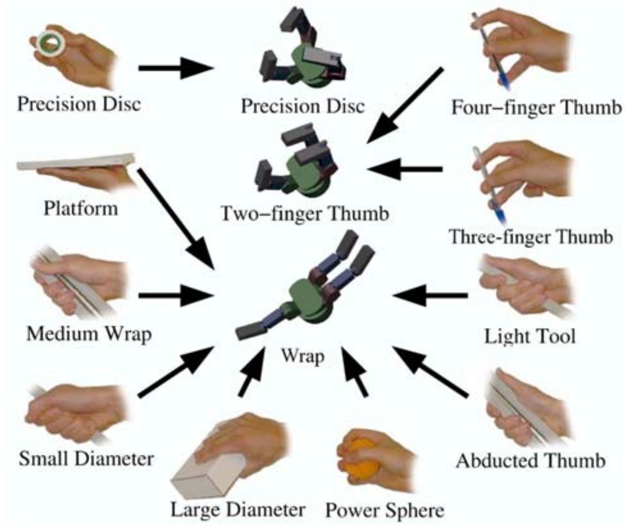


Fig. 1 Mapping a selection of grasps from Cutkosky's grasp hierarchy to three Barrett grasps. The robot hand configurations shown are the initial joint positions

generalized the existing grasp taxonomies [14]. Based on the above work and as described in previous work [7], the current grasp recognition system can recognize ten different grasp types. The human grasps are mapped to the robot as shown in Fig. 1. The grasps refer not only to hand postures, but to grasp execution schemes that include initial position, *approach vector*, and hand controller.

3 Grasp control

The grasp control algorithm has to be able to cope with occlusion and limited accuracy of the vision system. The controller presented here copes with the above without exploiting the wrist or arm motion. It is assumed that touch sensors capable of detecting the normal force are mounted to the distal links of each finger of the robot hand. This type of touch sensors are available at a low cost and are easy to mount on an existing robot hand as shown in previous work [15]. Considerations on different tactile sensors are put in to perspective in [13,16,28].

The hybrid force/position controller uses these tactile sensors to control the grasp force. Position control using joint encoders maintains the desired finger configuration and hence object position. Here, an effort is made to bring the object toward the center of hand during grasping. The need of this can be exemplified by the BarrettHand for which the grasp is typically of higher quality when all fingers have approximately the same closing angle, rather than when the object is far from the palm center. This behavior can be seen in the example task shown in Fig. 2. Here, the BarrettHand is modeled such that the two joint angles of each finger have a fixed relation. All control is performed using Matlab. Alternative low-level controllers have been investigated in e.g. [31].

To enable a more intuitive formulation of the controller—as opposed to decentralized control of reference trajectories and/or torques in joint space—a linear transform T is used to transform the *joint angles* q to more intuitive control variables $x = Tq$. If T is a diagonal matrix containing the finger lengths, x contains the finger tip position along the tip trajectory. Using a diagonal matrix T will hence allow for fingertip position control. Using a nondiagonal matrix T enables control of linear combinations of the joint angles q. Total closure—defined as the sum of the closing angle for all three fingers—for example, can be controlled using forces, positions, or any combination thereof. The joint space will be denoted q-space. The transformed space will be denoted x-space.

The choice of the matrix T is hence paramount to grasp controller behavior. Luckily, choosing T is quite straight forward assuming there exist an idea of how to control the hand. This idea may be to grasp the object using a certain force and hold the object close to the center of the palm. In this paper, a weighted sum of the contact forces, finger position difference (stability), off-set (centering), and finger spread is controlled (see Fig. 3 and [29] for more detail). The transform T used in this paper is

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 1 \\ 0 & 1/2 & 1/2 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}. \tag{1}$$

The control forces f (in x-space) are computed using a P-controller $f = De$ where D contains controller gains and e is an error vector. The joint torques (in q-space, used to actuate the hand in simulation) can now be computed as

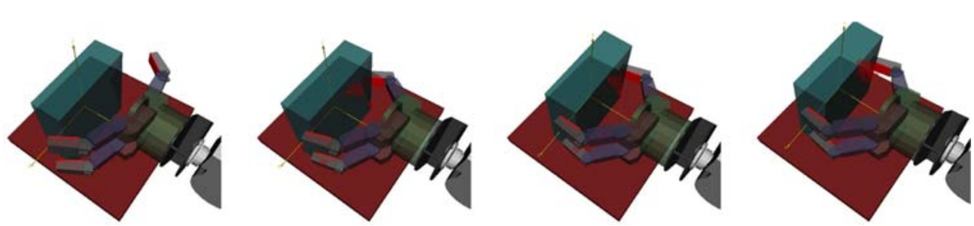


Fig. 2 Execution of a sample task where corrective movements are used to center the object

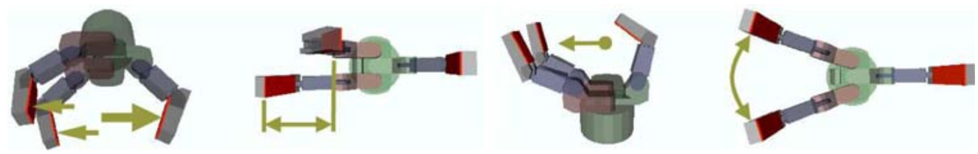


Fig. 3 Grasp control is performed using a transformation from joint angles q to linear combinations thereof $x = Tq$. Using a nondiagonal matrix T it is possible to control, for example, weighted grasp force, finger position difference (stability), off-set (centering), and finger spread. The distal links have thin tactile sensors mounted on them

$F = T^T f = T^T De$. As it is desired to perform position control for spread, stability, and centering and force control for the total closure, the error e is computed using the desired [des] values (in x-space) and the actual [act] sensor readings (of position and force) as

$$\begin{aligned}
e &= \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}^T \\
e_1 &= \begin{bmatrix} 1\,0\,0\,0 \end{bmatrix} e_x \\
e_2 &= \begin{bmatrix} 0\,1\,0\,0 \end{bmatrix} e_f \\
e_3 &= \begin{bmatrix} 0\,0\,1\,0 \end{bmatrix} e_x \\
e_4 &= \begin{bmatrix} 0\,0\,0\,1 \end{bmatrix} e_x \\
e_f &= f_{\text{des}} - f_{\text{act}} = f_{\text{des}} - T f_{\text{act}}^{\text{tactile sensors}} \\
e_x &= x_{\text{des}} - x_{\text{act}} = x_{\text{des}} - T q_{\text{act}}.
\end{aligned} \tag{2}$$

Fig. 4 *Top* the real objects. *Middle* the detailed object models. *Bottom* the simplified models, *object primitives*

4 Grasp learning

The object is first recognized using appearance (textural properties). When recognizing a known object, the object model can be quite detailed, but as a detailed object model is not always available, we perform training both on the primitive model and on the detailed object model (Sect. 1.1 and Fig. 4) and compare the results. Recent progress presented in [3] shows a promising method for retrieving shape primitives using vision, although the method is restricted to objects with uniform color.

While the grasp type is learnt from the demonstration and mapped to a robot hand grasp, learning of the approach vector is performed using a simple search technique where many different approach vectors are tested on the object. The approach vectors are generated by equidistant spacing (in spherical coordinates with the origin in the object center of mass and in Cartesian coordinates) or by using random displacement vectors. As previously mentioned, the training can be performed on either the primitive model or on the full object model. In the experiments, we have evaluated both methods.

Two approaches were used for learning. For *Power Grasps*, the hand is moved along an approach vector until or just before it contacts the object, and then the hand controller is engaged. For *Precision Grasps*, the approach is the same, but after contact with the object, the hand is retracted a distance d a certain number of times. After each retraction, the controller is engaged with the goal of reaching a fingertip (precision) grasp instead of a wrapping grasp.

For power grasps, the three parameters $(\theta,\ \phi,\ \psi)$ describing the approach direction and hand rotation are varied. The number of evaluated values for each variable is 9 for θ, 17 for ϕ, 9 for ψ, and for precision grasps the hand is retracted six times (d). For the precision grasps, the search space was hence 8,262 grasps, which required about an hour of training using kinematic simulation. For the power grasp simulations, 1,377 approach vectors were evaluated. The 5-s long grasping sequence is dynamically simulated in 120 s (Intel P4, 2.5 GHz, Linux). The quality measures for each grasp is stored in a *grasp experience* database.

4.1 Grasp quality measures

To evaluate grasps, the 6-D convex hull spanned by the forces and torques resistible by the grasp is analyzed using GraspIt!. The ϵ-L1 quality measure is the smallest maximum wrench the grasp can resist and is used for power grasps. For precision grasps, a grasp quality measure based on the volume of the convex hull was used, volume-L1. These grasp quality measures require full knowledge of the world and can thus only be used in simulation.

4.2 Grasp retrieval

At run-time, the robot retrieves the approach vector for the highest quality grasp from the grasp experience database. As the highest quality grasp is not necessarily the most robust with respect to position and model errors, the grasp should be chosen taking also those parameters into account (see Sect. 5.2). In a PbD scenario, the mapping from human to robot grasp is many-to-one, but if the robot acts autonomously, i.e. *explores* the environment and performs grasp on unknown objects, the grasp type is not defined and the best grasp can be chosen from among all possible grasps.

5 Experimental evaluation

This section provides experiments that demonstrate (1) robot grasping given the current state of the environment and the *grasp experience* database, and (2) how errors in pose estimation affect the grasp success. The objects were placed on a table [Fig. 5(left)]. Figure 5(right) shows the results of object recognition and pose estimation process. The human teacher, wearing a data-glove with magnetic trackers, moves an object. The move is recognized by the vision system and so is the grasp the teacher used. This information is used to generate a suitable robot grasp (grasp mapping) that controls the movement of the robot hand in the simulator.

Fig. 5 *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

5.1 Dynamic simulation

Grasping the rice box with a fingertip grasp was dynamically simulated using the controller from Sect. 3. Of the 1,377 approach vectors, 1,035 were automatically discarded, because the hand interfered with the table upon which the box is placed while approaching the object, or that the object was obviously out of reach. The remaining 342 initial robot hand positions were evaluated and resulted in 171 force closure grasps, 170 failed grasp attempts, and one simulation error. The top three hand initial positions and the resulting grasps are shown in Fig. 6. Some sample data from the third best simulation (Fig. 6c, f) is shown in Fig. 7. The desired grasping force is set to 5 N. A low-pass filter is used for the tactile sensor signal.

5.2 Introducing errors in the pose estimation

To evaluate the performance under imperfect pose estimation, we have simulated errors in pose estimation by providing an object pose with an offset. In the first simulation experiment, grasping the rice box with a fingertip grasp, the object was translated a certain distance in a *random* direction.

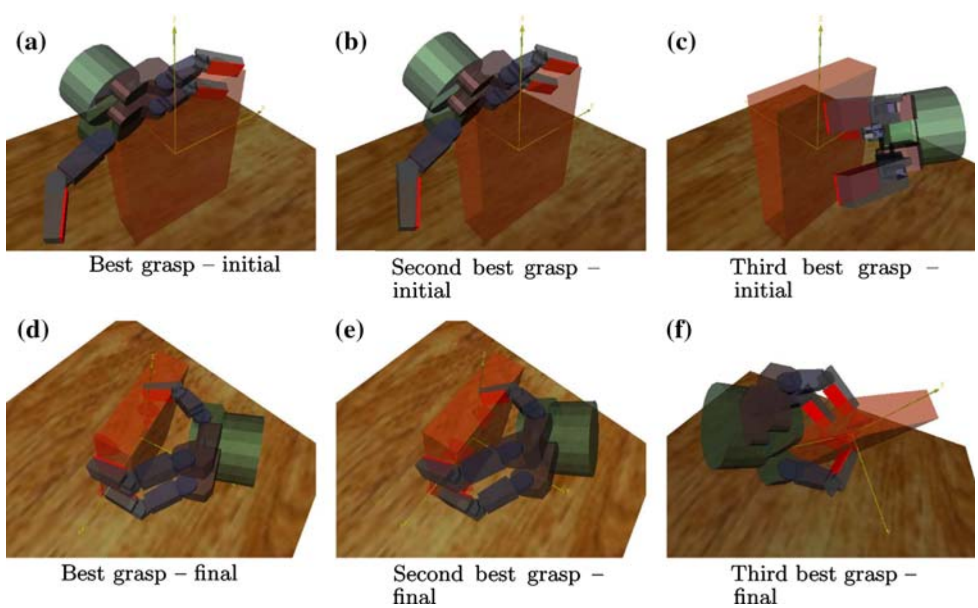
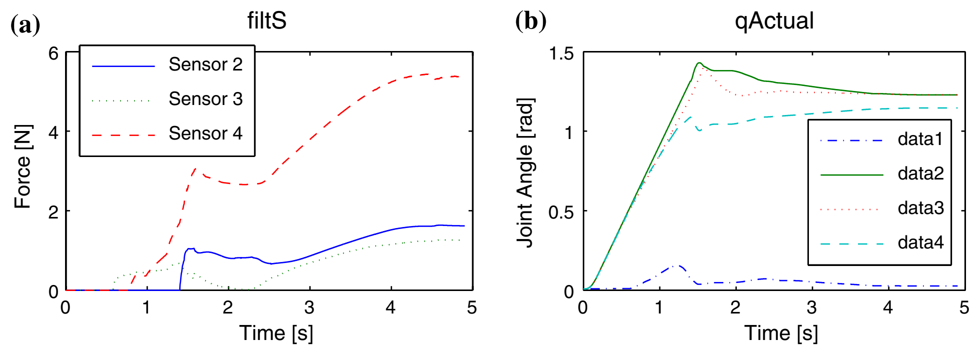


Fig. 6 The top three approach positions and the final grasps for fingertip grasping of the rice box. These results show that it is important to consider the dynamics when designing grasp execution schemes and for analyzing the grasp formation process. In several simulations, the fingers stop after contacting the box (as they should), but when the grasping force is increased, the box slides on the low friction proximal links and also on the resting surface until it comes in contact with the high friction tactile sensors

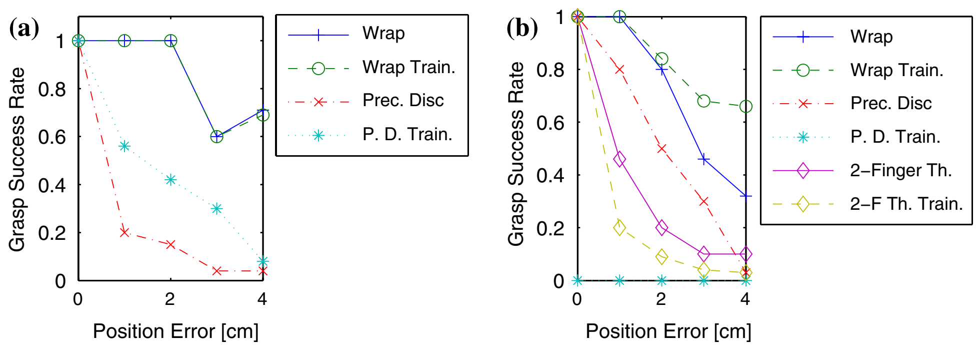
Fig. 7 Data logged from the grasp simulation in Fig. 6c, f. In the first 1.3 s, the fingers close under force control. The force at that time is used as the start value for the force controller that ramps the grasp force to 5 N. The joint angle values show that the joint angles are getting closer to equal with time



Tactile sensors - filtered

Joint angles

Fig. 8 The effect of position errors on the grasp success rate. For these results, the training and evaluation was performed using kinematic simulation only. *Prec. Disc* precision disc, *P. D. Train* precision disc trained, *2-Finger Th.* two-finger thumb, *2-F Th. Train* two-finger thumb trained



Grasping a bottle.

Grasping a mug.

As a result, the robot interpreted the situation as if the object was in another position than that for which the grasp was planned. This was repeated 50 times for five different vector lengths: 0, 1, 2, 3, and 4 cm. In total, there were 250 grasps from 201 positions. Figures 8 and 9 show the grasp success rates for various grasps and objects, under increasing error in position estimation. A grasp is considered successful if it results in force-closure.

For the second experiment, the scenario was grasping a bottle using a wrap grasp. In the initial position, the bottle was centered with respect to the palm and a fraction of a millimeter away. It was then repositioned at positions in a grid with 20 mm spacing and another grasp was performed. For this scenario, the position of the bottle does not need to be very accurate (see Fig. 10).

Using kinematic simulation, we have evaluated how an error in rotation estimate affects the grasp formation. As expected, for symmetric objects like the orange and the bottle, this type of error has no effect. Table 1 shows the rotation tolerance for different objects and grasp types. For grasping

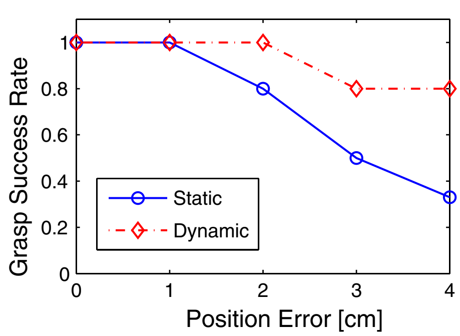


Fig. 9 The need for using dynamic simulation in grasp formation analysis is obvious. The grasp is the same as seen in Fig. 6c, f. (Because of some problems with the simulator, a limited number of samples were used in the evaluation of dynamic grasping. For the 0, 1, 2, 3, and 4 cm random displacement, the number of trials were 50, 14, 18, 18, and 12, respectively (instead of 50). Still, these samples were truly random and we believe that the number of trials is high enough to draw conclusions)

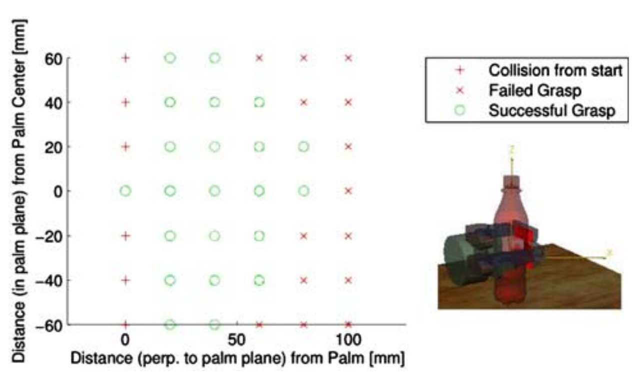


Fig. 10 Grasp success as a function of initial bottle position. Grasp success is here defined as reaching a force closure grasp. The $(x, y) = (0, 0)$ position is right in front of the palm with a fraction of a millimeter of space between the palm and bottle. The *inset* shows the final grasp from $(x, y) = (40, 40)$

Table 1 Rotation error tolerances

Object	Grasp type	Rotation error tolerance (degrees)
Zip disc box	Wrap	3
Rice box	Wrap	17
Mug	Wrap	12
Mug	Precision disc	0
Mug	Two finger thumb	6

the mug, the rotation estimation is absolutely crucial. Thus, this type of grasp should be avoided for this object.

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. Dynamic simulation and the controller previously outlined yields significantly better grasps than using purely kinematic simulation (Fig. 9). This is a motivation for continuing the investigations on the dynamics of the grasp formation process.

The bottle and the mug have been trained using both primitive and detailed models. Training using the primitive models does not decrease the grasp success rate by much in most cases. 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.

5.3 Discussion

The success rate of the presented system depends on the performance of four subparts: (1) object recognition, (2) grasp recognition, (3) pose estimation of the grasped object, and (4) grasp execution. As demonstrated in previous papers [7,8], 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 static environment may be considered close to perfect with respect to the first steps. As the object pose and possibly the object model is not perfectly known, some errors were introduced that indicate the needed precision in the pose estimation under certain conditions. Initial results suggest that for certain tasks, grasping is possible even when the object's position is not perfectly known.

If a high quality dynamic physical modeling is essential, for example when grasping compliant objects or for advanced contact models, other simulation tools than GraspIt! may be more suitable (see, e.g. [11]).

6 Conclusions

A framework for generating robot grasps based on object models, shape primitives, and/or human demonstration have been presented and evaluated. The focus lies on the choice of approach vector, which depends on the object's pose and grasp type. The approach vector is based on perceptional cues and on experience that some approach vectors will provide better tactile cues that result in stable grasps. Another issue considered is obtaining stable grasps under imperfect vision, something that has not been thoroughly investigated in the literature.

Simulating results were necessary for generating insight into the problem and for performing the statistical evaluation for the grasp experience, since (1) the world must be reset after each grasp attempt and (2) computing the grasp quality measure requires perfect world knowledge. The proposed strategies have been demonstrated in simulation using tactile feedback and hybrid force/position control of a robot hand. The functionality of the proposed framework for grasp scheme design has been shown by successfully reaching force closure grasps using a BarrettHand and dynamic simulation.

Future work includes further grasp execution scheme development and implementation. Furthermore, to ensure truly secure grasping outside the simulator, the grasping scheme must also comprise a grasp quality evaluation method that does not use information available in simulation only. Preferably, such a measure would also depend upon the task at hand.

The grasp experience database contains not only a record of success rates for different grasp controllers but also the object–hand relations during an experiment. In this way, we can specify under what conditions the learnt grasp strategy can be reproduced in new trials. The results of the experimental evaluation suggest that the outlined approach and tools can be of great use in robotic grasping, from learning by demonstration to robust object manipulation.

Acknowledgments This work was supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657, by the Knowledge Foundation through AASS at Örebro University, and by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems at KTH. Thanks to Johan Sommerfeld for helpful software.

References

1. Aleotti J, Caselli S, Reggiani M (2004) 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
2. Bicchi A, Kumar V (2000) Robotic grasping and contact: a review. In: Proceedings of the IEEE international conference on robotics and automation, ICRA'00, pp 348–353
3. Bley F, Schmirgel V, Kraiss K (2006) Mobile manipulation based on generic object knowledge. In: IEEE international symposium on robot and human interactive communication, RO-MAN'06
4. Borst C, Fischer M, Haidacher S, Liu H, Hirzinger G (2003) DLR hand II: experiments and experiences with an antropomorphic

hand. In: IEEE international conference on robotics and automation, vol 1, pp 702–707
5. Cutkosky M (1989) On grasp choice, grasp models and the desing of hands for manufacturing tasks. IEEE Trans Robot Autom 5(3):269–279
6. Ding D, Liu YH, Wang S (2000) Computing 3-d optimal formclosure grasps. In: Proceedings of the 2000 IEEE international conference on robotics and automation, pp 3573–3578
7. Ekvall S, Kragic D (2005) Grasp recognition for programming by demonstration. In: IEEE/RSJ IROS
8. Ekvall S, Kragic D (2005) Receptive field cooccurrence histograms for object detection. In: IEEE/RSJ IROS
9. Ekvall S, Kragic D (2006) Learning task models from multiple human demonstration. In: IEEE international symposium on robot and human interactive communication, RO-MAN'06
10. Ekvall S, Kragic D, Hoffmann F (2005) Object recognition and pose estimation using color cooccurrence histograms and geometric modeling. Image Vis Comput 23:943–955
11. Ferretti G, Magnani G, Rocco P, Viganò L (2006) Modelling and simulation of a gripper with dymola. Mathematical and Computer Modelling of Dynamical Systems
12. Friedrich H, Dillmann R, Rogalla O (1999) Interactive robot programming based on human demonstration and advice. In: Christensen H et al (eds) Sensor based intelligent robots, LNAI1724, pp 96–119
13. Howe RD (1994) Tactile sensing and control of robotic manipulation. Adv Robot 8(3):245–261
14. Iberall T (1997) Human prehension and dextrous robot hands. Int J Robot Res 16(3)
15. Kragic D, Crinier S, Brunn D, Christensen HI (2003) Vision and tactile sensing for real world tasks. In: Proceedings IEEE international conference on robotics and automation, ICRA'03, vol 1, pp 1545–1550
16. Lee MH, Nicholls HR (1999) Tactile sensing for mechatronics—a state of the art survey. Mechatronics 9(1):1–31
17. Massimino M, Sheridan T (1989) Variable force and visual feedback effects on teleoperator man/machine performance. In: Proceedings of NASA conference on space telerobotics
18. Miller AT, Allen P (2000) Graspit!: a versatile simulator for grasping analysis. In: Proceedings of the of the 2000 ASME international mechanical engineering congress and exposition
19. Miller AT, Knoop S, Allen PK, Christensen HI (2003) Automatic grasp planning using shape primitives. In: Proceedings of the IEEE international conference on robotics and automation, pp 1824–1829
20. Morales A, Chinellato E, Fagg AH, del Pobil A (2004) Using experience for assessing grasp reliability. Int J Hum Robot 1(4):671–691
21. Namiki A, Imai Y, Ishikawa M, Kaneko M (2003) Development of a high-speed multifingered hand system and its application to catching. In: IEEE/RSJ international conference on intelligent robots and systems, vol 3, pp 2666–2671
22. Napier J (1956) The prehensile movements of the human hand. J Bone Joint Surg 38((4):902–913
23. Platt RJ (2006) Learning and generalizing control-based grasping and manipulation skills. PhD thesis, University of Massachusetts
24. Platt R Jr, Fagg AH, Grupen RA (2003) Extending fingertip grasping to whole body grasping. In: Proceedings of the international conference on robotics and automation
25. Pollard NS (1994) Parallel methods for synthesizing whole-hand grasps from generalized prototypes. PhD thesis, Massachusetts Institute of Technology
26. Pollard NS (2004) Closure and quality equivalence for efficient synthesis of grasps from examples. Int J Robot Res 23(6):595–613
27. Skoglund A, Iliev B, Palm R (2008) A hand state approach to imitation with a next-state-planner for industrial manipulators. In: International conference on cognitive systems, Karlsruhe, Germany
28. Tegin J, Wikander J (2004) Tactile sensing in intelligent robotic manipulation—a review. Ind Robot 32(1):64–70
29. Tegin J, Wikander J (2006) A framework for grasp simulation and control in domestic environments. In: IFAC-Symposium on mechatronic systems, Heidelberg, Germany
30. Townsend W (2000) The BarrettHand grasper—programmably flexible part handling and assembly. Ind Robot Int J 27(3):181–188
31. Volpe R, Khosla P (1993) A theorethical and experimental investigation of explicit force control strategies for manipulators. IEEE Trans Autom Control 38(11):1634–1650