Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D4.1.2**
Title of the deliverable:	**Technical Report on learning attribute-action relations from visual and haptic feedback**

Contractual Date of Delivery to the CEC:	1.2.2008
Actual Date of Delivery to the CEC:	31.1.2008
Organisation name of lead contractor for this deliverable:	SDU

Author(s): Dirk Kraft, Norbert Krüger, Mila Popovic, Kai Welke, Tamim Asfour, Damir Omrčen, Aleš Ude, Kai Hübner, Danica Kragic, Alex Bierbaum, Alejandro Agostini, Florentin Wörgötter, Christopher Geib, Ron Petrick, Mark Steedman, Bernhard Hommel, Justus Piater, Rüdiger Dillmann
Participant(s): KTH, BCCN, AAU, JSI, UL, UEDIN, SDU, ULg, UniKarl

Work package contributing to the deliverable:	WP1,WP2,WP8
Nature:	R
Version:	Final
Total number of pages:	10
Start date of project:	1st Feb. 2006 **Duration:** 48 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)
Dissemination Level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

In this deliverable, we describe our work on attribute–action associations. We describe progress on tactile and haptic sensing, a number of initially predefined behaviours based on object-action associations as well as our work on refining such associations by learning. Finally we present a more theoretical work on using anticipation to extend the human body by tools.

Keyword list: Vision, Tactile Sensing, Object-Action Associations

Table of Contents

1. Introduction

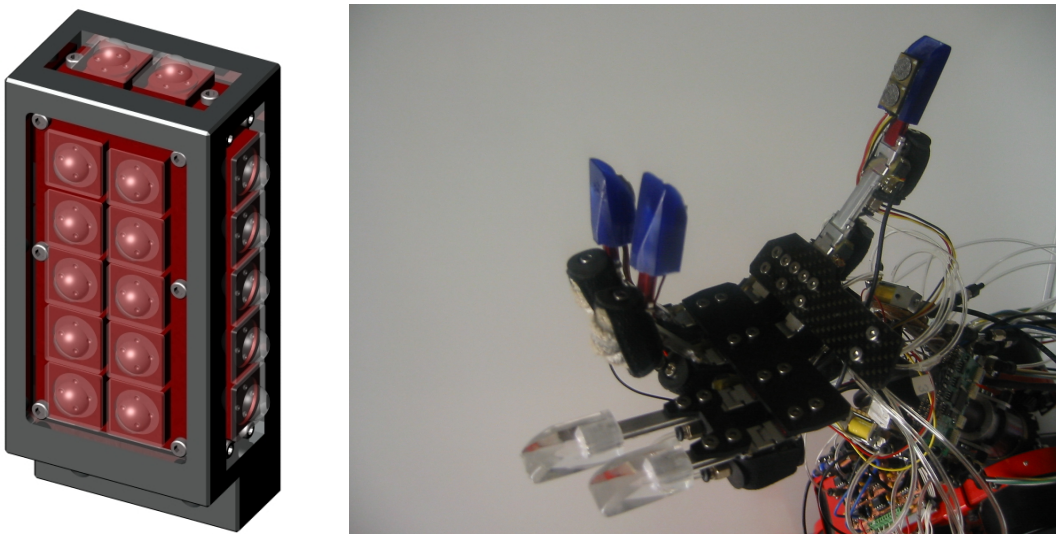
A summary of our work on the learning of attribute-action relations is described in this deliverable. Extended descriptions are given in a set of accepted [A, C] or submitted [G] publications as well as technical reports [B, D, E, F] which are attached.

In section 2, we briefly describe the further developed hardware set-up used at UniKarl and SDU in particular in terms of more sophisticated tactile sensing. By this set-up, we provide the basis for the multi-sensorial information that is needed and used in the context of the attribute–action relations being developed in particular in the context of WP4 and WP8. In section 3, we describe work on predefined part-action associations (in the following called 'reflexes') that are used in the bootstrapping process of the system (see also Deliverable D8.1.4). This work has been published in [A, C]. We have done first steps in the direction of learning more refined behaviour based on these reflexes. Most importantly, we are now able to generate large data sets with labelled data consisting of successful and un-successful grasps and associated feature combinations (see [F]). In section 4, the learning of object-action associations is described. As pointed out also in WP8, we see it as a relevant feature of our approach that learning is taking place on rather different levels of the processing hierarchy. In this context, we present work on learning on a very low level, where the consequences of a simple poking action on the pose of a rigid object become learned. We also give an example for learning within an embodied system which takes place on a symbolic level in terms of acquiring rules (see also WP5). A very relevant object in the context of PACO-PLUS is a tool since a tool requires to reason not only about the consequences of the robot's actions on objects but on the extension of the body by tools and the extended action options that come with that. In section 5, we describe our more preparational and theoretical work on the integration of tools into the concept of a body. The work described here is or will be relevant for the demos described in WP8 where they serve as sub-modules of the cognitive system we are aiming it.

2. Haptic and tactile Sensing

In the context of WP4, we mainly work within two hardware platforms, one at SDU and one at UniKarl. Both platforms deliver haptic and tactile feedback which is used to learn object-action relations. Although both platforms are being equipped with a robot and gripper, an (active) stereo system as well as force-torque sensors and tactile sensors based on micro-joystick technology, the SDU platform has a much lower degree of complexity since there is no active camera steering but a fixed camera robot setting. Also the SDU system has a very precise 6DoF industrial robot with a two finger gripper compared to a redundant pair of human like arms with a five finger hand used at UniKarl. The role of these two different platforms within PACO-PLUS are clearly defined. Because of the reduced complexity in the SDU system, it is easier to perform longer action sequences, to test complex architectures and to make use of or rely on more precise information for actions such as grasping and learning. However, research done initially in the SDU environment finally has to be transferred to the UniKarl platform as being done in the context of planning already.

On both platforms, progress on tactile and haptic sensing has been achieved over the last year. The SDU system is now equipped with a force-torque sensor at the wrist by which information about collisions can be detected and a withdraw action can be performed before too strong forces build up that would lead to an emergency stop or the destruction of objects. Furthermore, weight measurements as well as information about openness and closedness of objects can be detected haptically. In addition, after having evaluated the potential of the micro-joystick based technology for tactile sensing (see [D]), we have been building a prototype for a finger fully equipped with tactile sensors (see figure 1(a)). The five finger hand used at UniKarl is equipped with the very same tactile sensors as used in the SDU system (see figure 1(b)).



(a) (b)

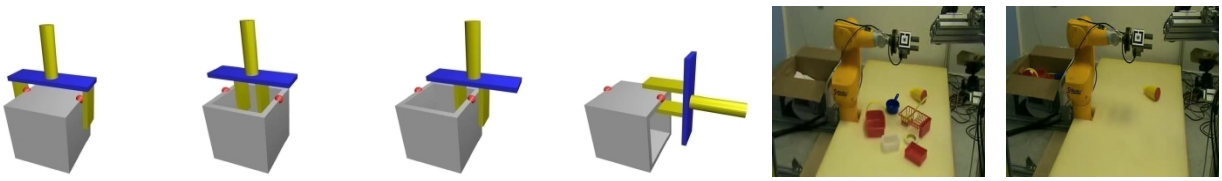
Figure 1: (a) Rendered design of the new finger developed at SDU. The finger is meant for a two finger gripper and is covered with sensors on five sides. (b) Humanoid robot at UniKarl.

3. Part-Action Association

We have worked on three different part–action associations, two being connected to contour structures (section 3.1) and one being connected to dense shape representations (section 3.2). This work has been described in more detail in [A, F, B, C].

3.1 A Grasping Reflex based on Co-planar and Circular Structures

We make use of two part-action association which are used as initial reflex-like behaviour in the complete system (see deliverable D8.1.4). One association (see figure 2(a)), which has already been worked on over the last year is based on co-planar pairs of contours. The potential of this grasping reflex has now been thoroughly evaluated in [F] where we could show, that even in complex scenes a large amount of objects can be grasped by these simple mechanisms (see figure 2(b,c)). To be able to deal with objects that have a 3D circle as a parts (as many objects have in a kitchen scenario), we also defined a part action association (see figure 3). Work on this, including the part extraction mechanism, is described in [B]. First steps in the area of improving these reflexes by learning are described in Deliverable D8.1.4.



(a) (b) (c)

Figure 2: Co-planar pairs of contours predict groups. (a) The four different elementary grasping actions defined based on a pair of co-planar groups. (b) Robot scene before the grasping procedure has been applied. (c) Scene after all graspable Objects have been removed by the system.

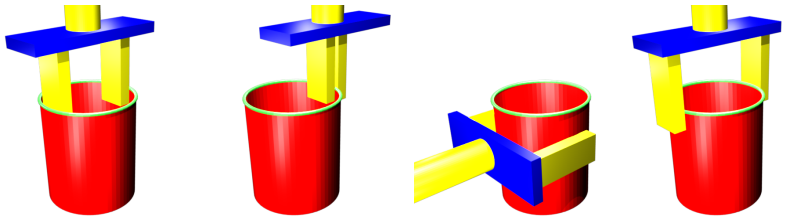


Figure 3: Grasp types. They are defined object centric and based on the upper circle.

3.2 Associations of Grasps to Objects using dense Representations

Robot grasping capabilities are necessary to actively execute tasks, modify scenarios and thereby reach versatile goals. These capabilities also include the generation of stable grasps to safely handle even objects unknown to the robot. In this paper, we follow the idea that the key to this ability is not primarily to select a grasp depending on the identification of a selected object, but on its shape. To approach this goal, [C] presents an algorithm that efficiently wraps given 3D data points of an object into primitive box shapes by a fit-and-split algorithm, based on Minimum Volume Bounding Boxes. Box shapes are not able to approximate arbitrary data in a precise manner, but it is shown that they give efficient clues for planning grasps on arbitrary objects, even more on object parts. Keeping in mind that it is not necessary to find the very best grasp, but one out of those that are stable, this seems reasonable. Additionally, the part-describing boxes allow for grasp semantics that might be mapped to boxes in the set, e.g. "approach the biggest part for a good grasp to stably move the object" or "approach the smallest part for a good grasp to show a most unoccluded object to a viewer." The description of an object by a shape-base part representation, which is claimed to be necessary for this kind of task-dependent grasping, is thereby made available (see figure 4). A more detailed description can be found in [C].

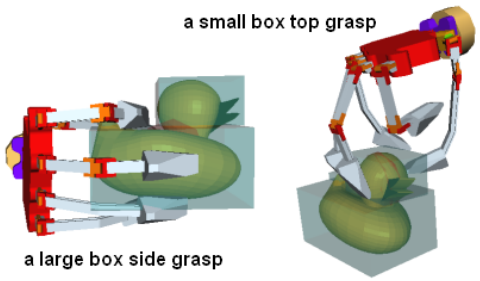


Figure 4: Minimum Volume Bounding Boxes and associated grasps

4. Learning Object-Action Relations

We addressed learning of object–action associations on a rather low level, where the effect of a poking action on an object becomes predicted (described in section 4.1) as well as on a rather symbolic level where abstract rules become learned (as described in section 4.2).

4.1 Learning Consequences of Poking Actions

Assuming that the robot can control its arm, it can start using it to act in the environment. The robot can make the initial object-action associations by observing the outcome of exploratory actions on objects. One of the most basic types of interaction with an object is poking, which can be defined as a short term pushing action. In the study in [E] we implemented an exploratory behaviour in which the robot randomly or systematically pushes an object from various sides and in different directions. The resulting object motions were identified and the robot used this data to associate the parameters of poking actions with the actual object movement. A neural network with two hidden layers was utilised to represent and learn these associations.

Finally, we showed how the acquired knowledge can be used to move the object in the desired direction by poking using feedback control (see figure 5).

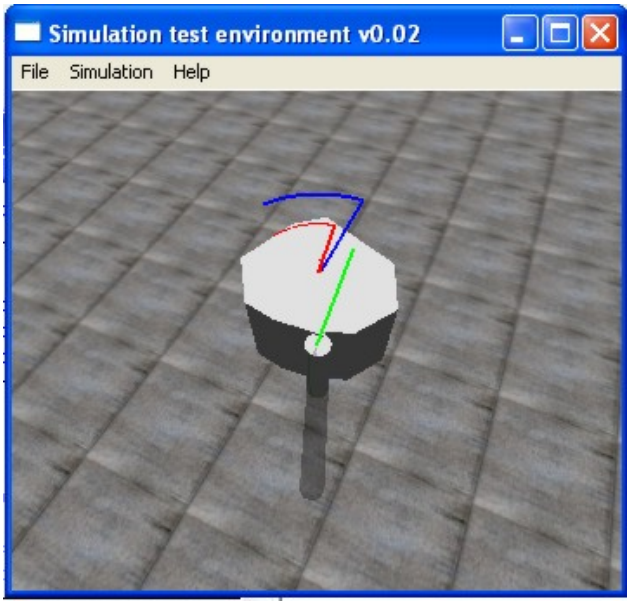


Figure 5: Execution of the pushing action (blue line desired object motion, red line actual object motion, green line pusher movement)

4.2 Learning Rules

In conjunction with WP5, we are developing methods for acquiring STRIPS/LDEC-like rules and plans defined as a set of preconditions, a sequence of one or more actions. The preconditions are a set of necessary conditions or perceptions that must be observed before the rule can be applied, and the expected outcome is a series of effects that will be obtained after the execution of the rule. The action sequence may consist of a single elementary action in the simplest rules (the cause-effect relation for that action) or a list of actions, each one expressed in turn as a cause-effect. The rules so formed are behaviours that can be memoized, Explanation-Based Learning-style, when the obtained sequence is frequent or expensive to calculate, and used as planning operators [6], [7], [9]. They are used in a decision making platform [1] in which the rules are acquired from natural human instructions about cause-effect relations in currently observed situations, minimising complicated instructions and explanations of long-run action sequences and complete world dynamics. Plans can also in principle be constructed automatically by the PKS planner, partly developed in PACO-PLUS under WP5. The process of learning these operators from interaction with the world is investigated elsewhere in WP5 [5].

5. Tools as Extensions of the Body

Embodied cognition suggests that complex cognitive traits can only arise when agents have a body situated in the world. When discussing the aspects of embodiment and situatedness from the perspective of linear systems theory one can treat bodies as dynamic, temporally variable entities, which can be extended (or curtailed) at their boundaries. In a set of experiments we could show how acting agents can, for example, actively extend their body for some time by incorporating predictably behaving parts of the world and how this affects the transfer functions. These studies suggest that primates have mastered this to a large degree increasingly splitting their world into predictable and unpredictable entities. We argue that this kind of temporary body extension may have been instrumental in paving the route for the development of higher cognitive complexity as it is reliably widening the cause-effect horizon about the actions of the agent.

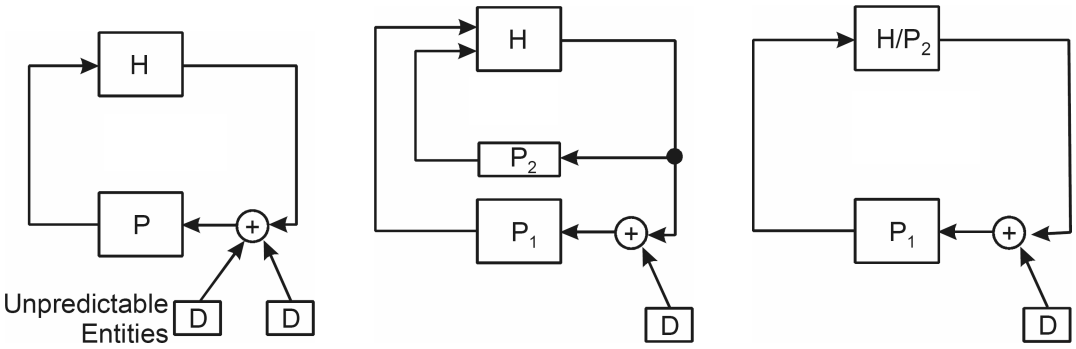


Figure 6: Systems theoretical representation of temporary embodiment.

Figure 6(a) depicts a situated agent (human) facing a few disturbances. In the process of grasping an object the human will — if successful — be able to make the grasped object fully (or at least very) predictable to her. Hence, an entity that had been a disturbance (of — say — her visual input space) will first become a predictable entity P2 (figure 6(b)), where the human will then be able to temporarily integrate this entity into her body (figure 6(c)). The division operation is mathematically fully correct in this case as long as we talk about linear transfer functions. An extention to more realistic non-linear cases would require a more elaborate mathematical treatment, which, however is not of interest for the main part of our discussion here. The remaining aspects P1 of the world cannot be integrated as they might, for example, be too unpredictable or too far away or from the agent's currently existing body. The idea that humans (and monkeys) indeed perform temporary bodily integration is supported by experimental results that over time cortical receptive fields are extended representing the tip of a stick, which a monkey had to use to obtain food for an prolonged period of time [8]. Hence a long duration, where the processes depicted in figure 6 had taken place, has in this case even led to a long-lasting plastic change of the nervous system of this agent (monkey).

The apparently strange notion of temporary bodily integration becomes much more digestible if one thinks of an advanced robot that has grasped a pair of pliers and can handle it now with high precision and dexterity. What would prevent us — the robot's designers — from using a few screws to permanently attach these pliers to the body of the robot this way making the temporary bodily integration a permanent one?

In the following we will describe a set of experiments performed with a simple industrial robot (Stäubli, Switzerland) demonstrating how the principle of temporary bodily integration can be implemented in a machine in a simple algorithmical way to provide some support to this idea.

To this end we assume a few things for our machine to be innate:

A A visual representation exists by which a scene can be decomposed into simple 3-D entities, which

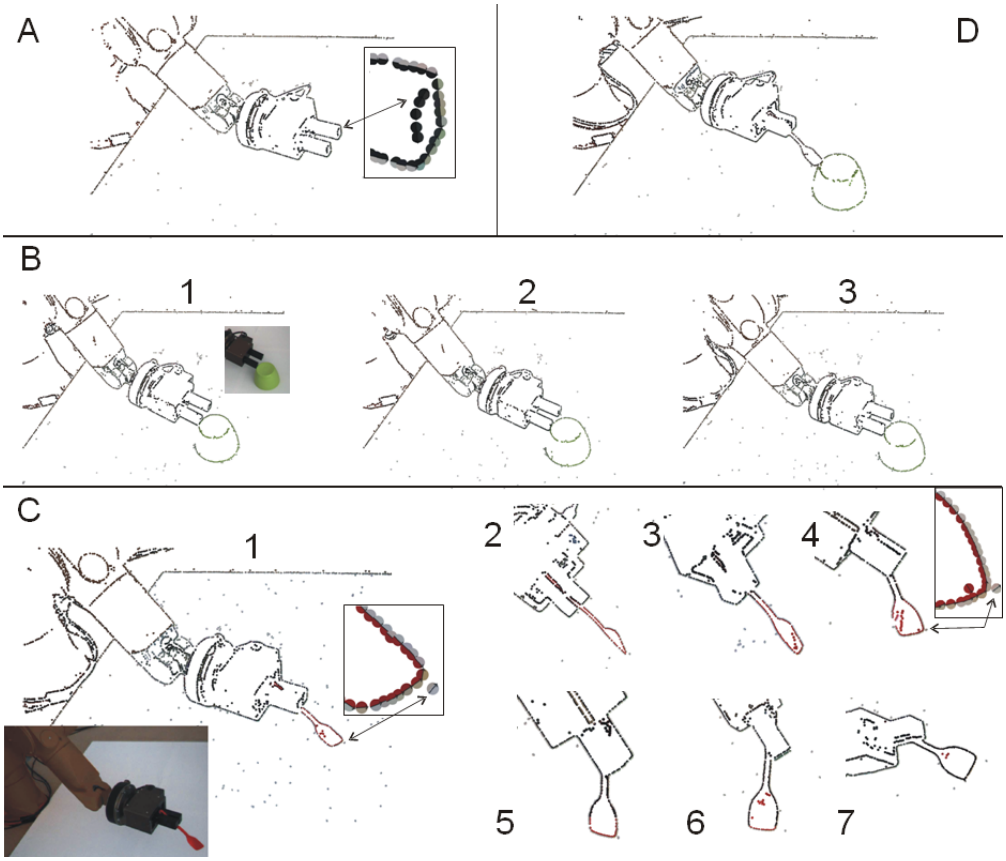


Figure 7: Temporary embodiment experiment (for explanation see text).

we call primitives (see figure 7A especially also the inset; for technical details see [4]). Notions of distance (metric) exist therein.

B The machine "knows" that coherently moving primitives belong together. This is known as the rigid body motion principle (see [2]) and corresponds to the prominent Gestalt law of Common Fate.

C Through this, the robot has learned about its own body (gripper). This can be achieved by a purely correlation based learning process where the robot has learned to associate coherent motion in the visual field to the fact that there has been a motor command, which the machine has used to perform a movement. We assume that the process of knowing its own body is basically completed; but that this process keeps on running in the background to safeguard against incompleteness and errors in the body representation.

D The machine can move its arm and it has also a certain drive to move its arm around (without which nothing would ever happen!)

E The machine can push things around by making (visually measured) contact to entities in the scene, which do not belong to the machine. Measurement relies again on the 3-D primitives for which the concept of distance exist.

F A grasping reflex can also be performed with some success, triggered by certain geometrical constellations between primitives from the world ([A]). **It can feel a successful grasp (haptic sensors) and it knows that it cannot perform another grasp without first letting go. Like babies, it, however, rather likes to hold on to a grasped entity. After some longer time it might however get bored and then it releases the object (also similar to small children).**

G It has an exploration drive by which it will first try to grasp a thing and if this fails (measured by the haptic sensors at the hands), it will try to push it instead. This exploration is triggered by novelty and will start as soon as something new (new primitives) are discovered in the scene.

These rather basic sub-procedural components are enough to drive the required process. Figure 7A shows the body-representation of the robot as viewed by itself. All black-grey[1] primitives have been learned earlier (process C) to belong to its body. In the following we will for simplicity use the primitive type "black-gray" like a mental concept to graphically depict if a primitive is deemed to belong to the robot. If an object enters the visual scene the robot will try to grasp it (process G). If unsuccessful it will push it around (process E). This is shown for a not-graspable, upside down, green cup in figure 7B, where three movement stages are shown (figure 7B1-B3). If a grasp is successful (figure 7C), it will move the object (process D) like the spoon in figure 7C, where we show seven snapshots of movement stages. At first it will realize that the object is represented by many primitives which belong together (process B). This, we had at some point called "Birth of an Object" as it represents a step where the physical "object-ness" of otherwise purely visual entities (the primitives) can be ascertained ([3]). If the machine does not accidentally drop the object but instead moves it for a longer time it will realize that the movement of these primitives will (albeit in a complicated geometrical way) be related to its own motor actions (process C). As it does not know better it will update its body-image based on this sensor-motor correlation and extend it to now include the coherently and predictively moving object (process C). This is shown in figure 7C by the gradual spread of the black-grey primitives along the spoon until the whole spoon is being re-coloured. Again we emphasise that this is just a graphical representation of the spreading inclusion of the spoon into the body image of the robot. If a new entity will enter the visual field now, sub-process G is triggered again. It feels reluctant to let go (process F) and, thus, another grasp is inhibited (also F), hence sub-processes G,E will lead to a pushing action now (figure 7D). As a consequence this agent, based on very primitive sub-processes, begins to perform an interaction between a very simple "tool" that extends its body (until it drops it) and the world.

Figure 7 shows the complete experiment as performed with our robot. Clearly there are many more rather technical details that we had to take care of until the robot actually could do all this (see [A] as well as [3] for details), but the complete sequence as such does not require an other component beyond those (A-G) listed above.

6. Links to other Workpackages

Deliverable D4.1.2 is linked to and makes use of work made in a number of workpackages. It is linked to the software and hardware integration issues dealt with in WP1. In Deliverable D8.1.4 a number of sub-modules are used that have been developed in WP4, most notably the two grasping reflexes and the integration with the higher level planning system (see WP5).

Attached Papers

[A] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early reactive grasping with second order 3D feature relations. In S. Lee, I. Hong Suh, and M. Sang Kim, editors, *Recent Progress in Robotics; ViableRobotic Service to Human, selected papers from ICAR'07*. Springer-Verlag Lecture Notes in Control and Information Sciences (LNCIS), 2007.

[1] Note, primitives at an edge are always showing two colours, one for the inside, the other for the outside. Here the robot appears dark (black) and the background brighter (grey) leading to a black-grey primitive.

[B] E. Başeski, D. Kraft, and N. Krüger. A hierarchical 3D circle detection algorithm applied in a grasping scenario. Technical Report 2008 – 2, Robotics Group, Maersk Institute, University of Southern Denmark, 2008.

[C] K. Huebner, St. Ruthotto, and D. Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *IEEE Int. Conf on Robotics and Automation (ICRA),*, 2008. (accepted).

[D] M. Kjærgaard, D. Kraft, H. Petersen, N. Krüger, A. Bierbaum, T. Asfour, and R. Dillmann. Tactile object exploration using cursor navigation sensors. Technical Report 2008 – 1, Robotics Group, Maersk Institute, University of Southern Denmark, 2008.

[E] D. Omrčen and A. Ude. Learning primitive motions for robot poking by exploration. Technical report, Jožef Stefan Institute, 2008.

[F] M. Popović. An early grasping reflex in a cognitive robot vision system. Master's thesis, University of Southern Denmark, 2008.

[G] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr. Cognitive agents - A procedural perspective relying on "predictability". 2008. (submitted).

References

[1] A. Agostini, E. Celaya, C. Torras, and F. Wörgötter. Action rule induction from cause-effect pairs learned through robot-teacher interaction. In *International Conference on Cognitive Systems (CogSys)*, Karlsruhe, Germany, 2008. (submitted to).

[2] O.D. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[3] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger. Birth of the object: Detection of objectness and extraction of object shape through object action complexes. *Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics*, 2008. (accepted).

[4] N. Krüger, M. Lappe, and F. Wörgötter. Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.

[5] K. Mourao, R. Petrick, and M. Steedman. Using kernel perceptrons to learn action effects for planning. In *International Conference on Cognitive Systems (CogSys)*, 2008. (submitted to).

[6] M. Newton and J. Levine. Evolving macro-actions for planning. In *International Conference on Automated Planning and Scheduling*, Providence, Rhode Island, USA, 2007.

[7] M. Nicolescu and M. Mataric. A hierarchical architecture for behavior-based robots. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 227–233, Bolgna, Italy, 2002.

[8] S. Obayashi, M. Tanaka, and A. Iriki. Subjective image of invisible hand coded by monkey intraparietal neurons. *NeuroReport*, 11:3499–3505, 2000.

[9] D. Rao, Z. Jiang, and Y. Jiang. Learning activation rules for derived predicates from plan examples. In *International Conference on Automated Planning and Scheduling*, Providence, Rhode Island, USA, 2007.

Early Reactive Grasping with Second Order 3D Feature Relations

Daniel Aarno, Johan Sommerfeld, Danica Kragic[1], Nicolas Pugeault[2], Sinan Kalkan, Florentin Wörgötter[3], Dirk Kraft and Norbert Krüger[4]

[1] Centre for Autonomous Systems, Computational Vision and Active Perception, School of Computer Science and Communication, Royal Institute of Technology, 10044 Stockholm, Sweden, {bishop, johansom, dani}@kth.se
[2] School of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, United Kingdom, npugeaul@inf.ed.ac.uk
[3] Bernstein Center for Computational Neuroscience, University of Goettingen, Bunsenstr. 10, 37073 Goettingen, Germany, {sinan, worgott}@bccn-goettingen.de
[4] The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark, {kraft, norbert}@mmmi.sdu.dk

Summary. One of the main challenges in the field of robotics is to make robots ubiquitous. To intelligently interact with the world, such robots need to understand the environment and situations around them and react appropriately, they need context-awareness. But how to equip robots with capabilities of gathering and interpreting the necessary information for novel tasks through interaction with the environment and by providing some minimal knowledge in advance? This has been a longterm question and one of the main drives in the field of cognitive system development.

The main idea behind the work presented in this paper is that the robot should, like a human infant, learn about objects by interacting with them, forming representations of the objects and their categories that are grounded in its embodiment. For this purpose, we study an early learning of object grasping process where the agent, based on a set of innate reflexes and knowledge about its embodiment. We stress out that this is not the work on grasping, it is a system that interacts with the environment based on relations of 3D visual features generated trough a stereo vision system. We show how geometry, appearance and spatial relations between the features can guide early reactive grasping which can later on be used in a more purposive manner when interacting with the environment.

1 Introduction

For a robot that has to perform tasks in a human environment, it is necessary to be able to learn about objects and object categories. It has been recognized recently that grounding in the embodiment of a robot, as-well as continuous learning is required to facilitate learning of objects and object categories [21, 4]. The idea is that robots will not be able to form useful categories or object representations by only being a passive observer of its environment. Rather

a robot should, like a human infant, learn about objects by interacting with them, forming representations of the objects and their categories that are grounded in its embodiment.

Central to the approach are three almost axiomatic assumptions, which are strongly correlated. These also represent the building blocks of our approach toward creating a cognitive artificial agent:

- Objects and Actions are inseparably intertwined; Entities (”things”) in the world of a robot (or human) will only become semantically useful ”objects” through the action that the agent can/will perform on them. This forms so-called Object-Action Complexes (named OACs) which are the building blocks of cognition.
- Cognition is based on recurrent processes involving nested feedback loops operating on, contextualizing and reinterpreting object-action complexes. This is done through actively closing the perception-action cycle.
- A unified measure of success and progress can be obtained through minimization of contingencies which an artificial cognitive system experiences while interacting with the environment or other agents, given the drives of the system.

To demonstrate the feasibility of our approach, we aim at building a robot system that step by step develop increasingly advanced cognitive capabilities. In this paper, we demonstrate our initial efforts towards this goal by designing a scenario for manipulation and grasping of objects.

One of the most basic interactions that can occur between a robot and an object is for the robot to push the object, i.e. to simply make a physical contact. Already at this stage, the robot should be able to form two categories: physical and non-physical objects, where a physical object is categorized by the fact that interaction forces occur. A higher level interaction between the robot and an object would exist if the robot was able to *grasp* the object. In this case, the robot would gain actual physical control over the object and having the possibility to perform controlled actions on it, such as examining it from other angles, weighing it, placing it etc. Information obtained during this interaction can then be used to update the robots representations about objects and the world. Furthermore, the successfully performed grasps can be used as ground truth for future grasp refinement, [4].

In this paper, we are interested in investigating an initial “reflex-like” grasping strategy that will form a basis for a cognitive robot system that, at the first stage, acquires knowledge of objects and object categories and is able to further refine its grasping behavior by incorporating the gained object knowledge, [1]. The grasping strategy does not require *a-priori* object knowledge, and it can be adopted for a large class of objects. The proposed reflex-like grasping strategy is based on second order relations of multi-modal visual features descriptors, called *spatial primitives*, that represent object’s geometric information, e.g. 3D pose (position and orientation) as well as its appearance information, e.g. color and contrast transition etc. [9], see Fig. 1. Co–planar tuples of the spatial primitives allow for the definition of a plane that can be associated to a grasp hypothesis. In addition, these local descriptors are part of semi-global collinear groups [18]. Furthermore, the color information (by defining co–colority in addition to co–planarity of primitive pairs) can be used to further improve the definition of grasp hypotheses. In this paper, we employ the structural richness of the descriptors in terms of their geometry and appearance as well as the structural relations co–linearity, co–planarity and co–colority to derive a set of grasping options from a stereo image.

We note that the purpose of this work is not to develop yet another grasping strategy for a specific setting, but rather to provide low-level grasping reflexes that can be used to generate successful grasps on arbitrary objects. These grasping reflexes are part of a larger framework on cognitive robotics where a robot is equipped only with a set of innate grasps which are used

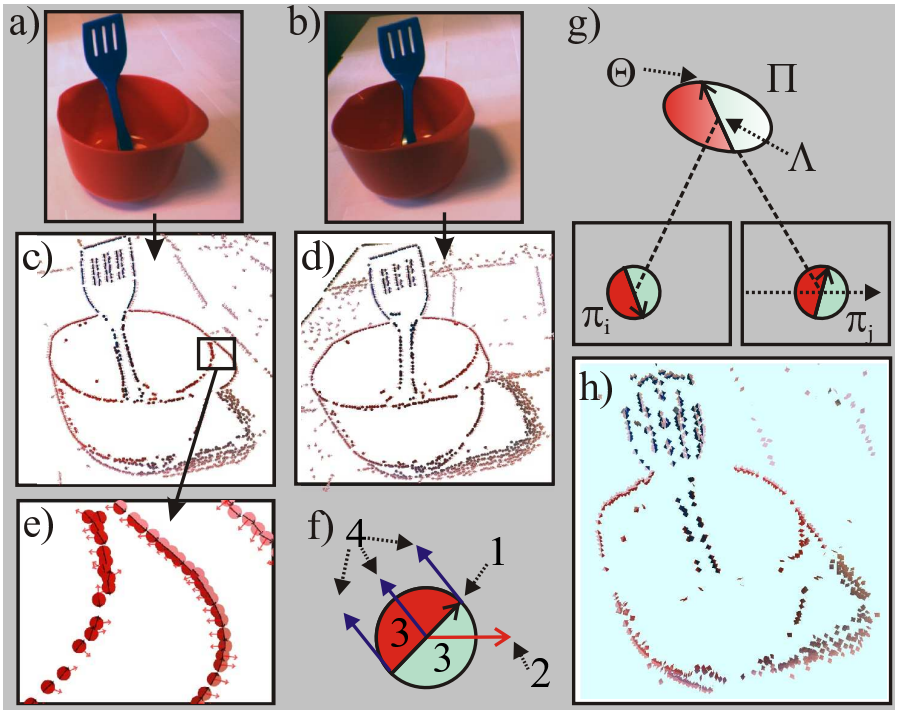


Fig. 1. Illustration of the vision module. a) and b) shows the images captured by the left and right cameras (respectively); c) and d) show the primitives extracted from these two images; in e) a detail of the primitive extraction is shown; f) illustrates the schematic representation of a primitive, where 1. represents the orientation, 2. the phase, 3. the color and 4. the optical flow. g) from a stereo–pair of primitives (π_i, π_j) we reconstruct a 3D primitive Π, with a position in space Λ and an orientation Θ; h) shows the resulting 3D primitives reconstructed for this scenario.

to develop more complex object manipulation abilities through interaction and reinforcement so that 1) more complex feature relations become associated to more precise and successful grasps, and 2) object knowledge becomes acquired and used to further refine the grasping process. We also have to stress out that no scene segmentation is performed, since the system does not even have a concept of an object to start with. In short, the contributions of our work are the generation of a set of grasp suggestions on unknown objects based on visual feedback, grouping of visual primitives for decreasing the size of the grasps and evaluation of grasps using the GraspIt! environment, [11].

In this work, "kitchen-type" objects such as cups, glasses, bowls and various kitchen utensils are considered. However, our algorithm is not designed for specific object classes but can be applied for any rigid object that can be described by edge–like structures.

This paper is organized as follows. In Section 2, we shortly review the related work and in Section 3 give a general overview of the system. Details about extraction of spatial primitives are presented in Section 4 and elementary grasping actions defined in Section 5. Results of the

experimental evaluation are summarized in Section 9 and plans for future research outlined in Section 10.

2 Related Work

The idea to learn or refine grasping strategies is not new. Kamon *et al.* combined heuristic methods with learning algorithms to learn how to select good grasps [7]. Rössler *et al.* used two levels of learners to learn local and global grasp criteria [19], where the local learner learns about the local structure of an object and the global learner learns which of the possible local grasps are best given the object.

There has been a large amount of work presented in the area of robotic grasping during the last two decades [2]. However, much of this work has been dealing with analytical methods where the shape of the objects being grasped is known *a-priori*. This work, referred to as *analytical methods*, has focused primarily on computing grasp stability based on force and form-closure properties or contact-level grasps synthesis based on finding a fixed number of contact locations with no regard to hand geometry, [2],[3]. This problem is important and difficult mainly because of the high number of DOFs involved in grasping arbitrary objects with complex hands. Another important research area is grasp planning without detailed object models where sensor information such as computational vision is used to extract relevant features in order to compute suitable grasps, [5, 20, 14]. In this paper, we denote this approach as *sensor-driven*.

Related to our work, we have to mention systems that deal with automatic grasp synthesis and planning, [12],[17],[13],[15]. This work concentrates on automatic generation of stable grasps given assumptions about the shape of the object and robot hand kinematics. Example of assumptions may be that the full and exact pose of the object is known in combination with its (approximate) shape, [12]. Another common assumption is that the outer contour of the object can be extracted and a planar grasp applied, [13]. 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 [16],[12]. [16] studies methods for adapting a given prototype grasp of one object to another object. The method proposed in [12] presents a system for automatic grasp planning for a Barrett hand [6] 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.

One difference between the analytical and sensor-driven approaches is that the former tend to use complex hands with many DOFs, while the latter use simple ones such as parallel yaw-grippers. One reason for this is that if the reconstruction of the object's shape is not very accurate, using a complex gripping device does not necessarily facilitate grasping performance. For sensor-driven approaches it is also very common to perform only planar grasps where all the contacts between the fingers and the object are confined to a plane. As an example, objects are placed on a table and grasped from above. This simplifies both the vision problem, since only the outer boundary of the object in the image plane has to be estimated, as well as the grasp planning by constraining the search space.

The main differences of our work compared to the abovementioned work are the following:

- We rely on 3D information based on three dimensional primitives extracted online. This allows us to compute arbitrary grasping directions compared to only planar grasps considered in, e.g. [13].

- The structural richness of the primitives (geometric and appearance based information, collinear grouping) allows for an efficient reduction of grasping hypotheses while keeping relevant ones.
- Our system focuses on generating a certain percentage of successful grasps on arbitrary objects rather than high quality grasps on a constrained set of objects. We will show that with our representations we are able to extract a sufficient number of successful grasping options to be used as initiator of learning schemes aiming at more sophisticated grasping strategies.

3 System Overview

The work presented in this paper serves as a building block for the development of a cognitive robot system. The robot platform considered is comprised of a set of sensors and actuators. The minimum requirements necessary to realize the work presented in this paper is that the sensors are able to deliver a set of visual primitives (section 4) and the configuration of the actuators. The required actuator is a manipulator, comprised of a robotic arm and a gripper device. In this context the term sensor is not necessarily related to a real physical sensing device, but rather an abstract measurement delivered to the system, possibly after performing computations on data sampled from a physical sensor.

The complete system is outlined in Fig. 2. In this paper we are interested in developing grasping reflexes. A grasping reflex is triggered by the vision system. The vision system continuously computes the spatial primitives described in section 4 which are feed as sensor input to the set of reflexes and to the cognitives system. If the grasping reflex has not been inhibited by the cognitive system and the sensor stimuli is strong enough, i.e. there are sufficiently many spatial primitives visible, the grasping reflex is performed. This reflex behavior computes a set of possible grasps and tries to perform them. Each grasp evaluated results in a reinforcement signal which can be used by the cognitive system to update its representation of the world. The following two sections describe the spatial primitives and the rules for generating the grasping actions.

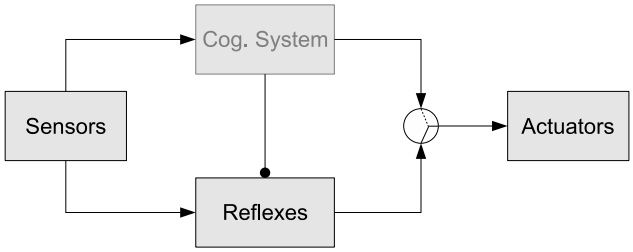


Fig. 2. System overview

4 Spatial Primitives

The image processing used in this paper is based on multi-modal visual primitives [10, 9, 18]. First, 2D primitives are extracted sparsely at points of interest in the image (in this case con-

tours) and encode the value of different visual operators (hereby referred to as *visual modalities*) such as local orientation, phase, color (on each side of the contour) and optical flow (see Fig. 1.d, 1.e and 1.f). In a second step, the 2D primitives become extended to the spatial primitives used in this work. After finding correspondences between primitives in the left and right image, we reconstruct a spatial primitive, (see Fig. 1.g) that has the following components, (for details see [8, 18]):

$$\Pi = \{\Lambda, \Theta, \Omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)\},$$

where Λ is the 3D position; Θ is the 3D orientation; Ω is the phase (i.e., contrast transition); and, $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color of the spatial primitive, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$).

The sparseness of the primitives allows to formulate three *relations* between primitives that are crucial in our context:

- *Co–planarity:*
  Two spatial primitives Π_i and Π_j are co–planar iff their orientation vectors lie on the same plane, i.e.:
  $$cop(\Pi_i, \Pi_j) = 1 - |\mathbf{proj}_{\Theta_j \times \mathbf{v}_{ij}}(\Theta_i \times \mathbf{v}_{ij})|,$$
  where $\mathbf{v}_{ij}$ is defined as the vector $(\Lambda_i - \Lambda_j)$, and $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$ is defined as:
  $$\mathbf{proj}_{\mathbf{u}}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2} \mathbf{u}. \quad (1)$$
  The co–planarity relation is illustrated in Fig. 3(a).
- *Collinear grouping (i.e., collinearity):*
  Two spatial primitives Π_i and Π_j are collinear (i.e., part of the same group) iff they are part of the same contour. Due to uncertainty in 3D reconstruction process, in this work, the collinearity of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. We define the collinearity of two 2D primitives π_i and π_j as:
  $$col(\pi_i, \pi_j) = 1 - \left| sin\left(\frac{|\alpha_i| + |\alpha_j|}{2} \right) \right|,$$
  where α_i and α_j are as shown in Fig. 3(b), see [18] for more details on collinearity.
- *Co–colority:* Two spatial primitives Π_i and Π_j are co–color iff their parts that face each other have the same color. In the same way as collinearity, co–colority of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. We define the co–colority of two 2D primitives π_i and π_j as:
  $$coc(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j),$$
  where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the RGB representation of the colors of the parts of the primitives π_i and π_j that face each other; and, $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is Euclidean distance between RGB values of the colors $\mathbf{c}_i$ and $\mathbf{c}_j$. In Fig. 3(c), a pair of co–color and not co–color primitives are shown.

Co–planarity in combination with the 3D position allows for the definition of a grasping pose; Collinearity and co–colority allows for the reduction of grasping hypotheses. The use of the relations in the grasping context is shown in Fig. 4.

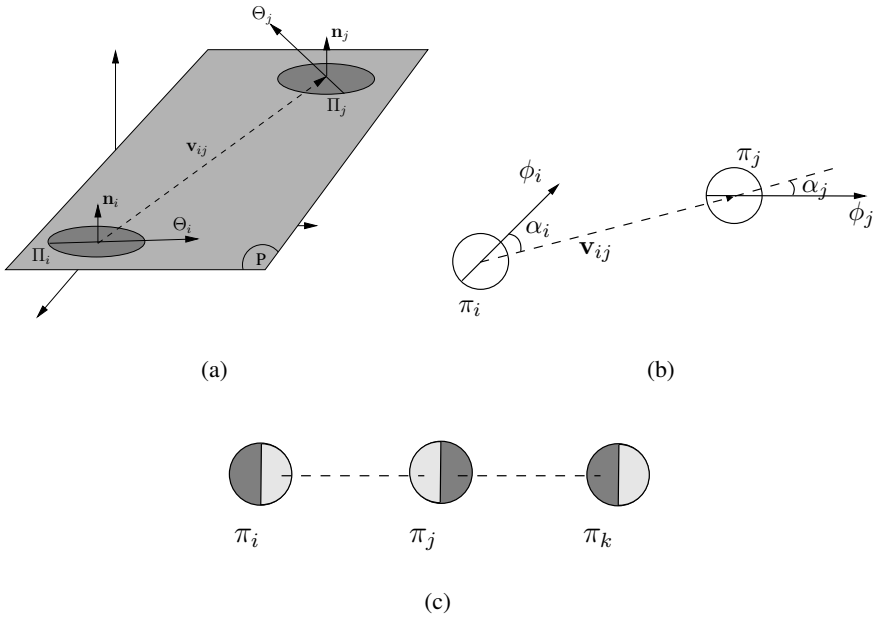


Fig. 3. Illustration of the relations between a pair of primitives. (a) Co–planarity of two 3D primitives Π_i and Π_j. (c) Co–colority of three 2D primitives π_i, π_j and π_k. In this case, π_i and π_j are cocolor, so are π_i and π_k; however, π_j and π_k are not cocolor. (b) Collinearity of two 2D primitives π_i and π_j.

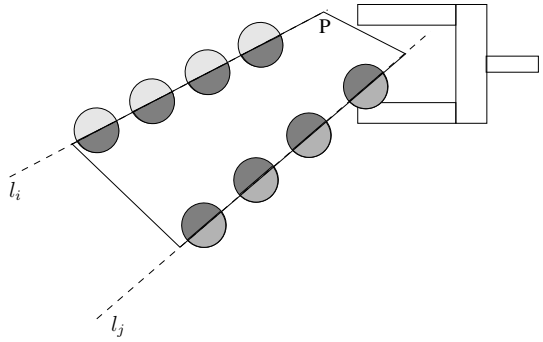
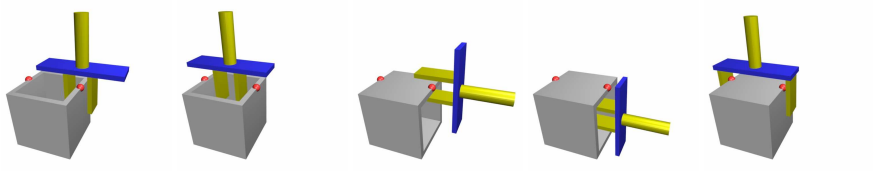


Fig. 4. A set of spatial primitives on two different contours l_i and l_j that have co–planarity, co–colority and collinearity relations; a plane P defined by the co–planarity of the spatial primitives and and an example grasp suggested by the plane.



(a) EGA1 (b) EGA2 (c) EGA3 (d) EGA4 (e) EGA5

Fig. 5. Elementary grasping actions, EGAs.

5 Elementary Grasping Actions

Coplanar relationships between visual primitives suggests different graspable planes. Fig. 4 shows a set of spatial primitives on two different contours l_i and l_j with co–planarity, co–colority and collinearity relations.

Five elementary grasping actions (EGA) will be considered as shown in Fig. 5. EGA1 is a "pinch" grasp on a thin edge like structure with approach direction along the surface normal of the plane spanned by the primitives. EGA2 is an "inverted" grasp using the inside of two edges with approach along the surface normal. EGA3 is a "pinch" grasp on a single edge with approach direction perpendicular to the surface normal. EGA4 is similar to EGA2 but its approach direction is perpendicular to the surface normal. Also it tries to go in "below" one of the primitives. EGA5 is wide grasp making contact on two separate edges with approach direction along the surface normal.

The EGAs will be parameterized by their final pose (position and orientation) and the initial gripper configuration. For the simple parallel jaw gripper, an EGA will thus be defined by seven parameters: EGA$(x,y,z,\gamma,\beta,\alpha,\delta)$ where $\mathbf{p} = [x,y,z]$ is the position of the gripper "center" according to Fig. 6; γ, β, α are the roll, pitch and yaw angles of the vector $\mathbf{n}$; and δ is the gripper configuration, see Fig. 6. Note that the gripper "center" is placed in the "middle" of the gripper.

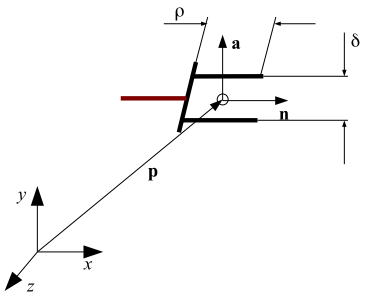


Fig. 6. Parameterization of EGAs.

The main motivation for choosing these grasps is that they represent the simplest possible two fingered grasps humans commonly use. The result of applying the EGAs can be evaluated to provide a reinforcement signal to the system. The number of possible outcomes of each of the EGAs are different and will be explained below.

For all of the EGAs the possibility of an *early failure* exists. That is, the EGA fails before reaching the target configuration. This will result in a reinforcement R_{fe}. Furthermore, it is possible for all EGAs to fail a grasping procedure.

For EGA1, EGA3 and EGA5, a failed grasp can be detected by the fact that the gripper is completely closed. This situation will result in a reinforcement R_{fl}.

For EGA1 and EGA3, the expected grasp is a pinch type grasp, i.e. narrow. Therefore, they can also "fail" if the gripper comes to a halt too early, that is $\delta > \Delta_{min}$. This will result in a reinforcement R_{ft}.

EGA2 fails if the gripper is fully opened, meaning that no contact was made with the object. This gives a reinforcement R_{fh}.

To detect failure of EGA4, a tactile sensor is required on the side of the "fingers". If, after positioning and opening the gripper, there is no contact between the object and the tactile sensor, the EGA has failed. This results in a reinforcement R_{fc}.

If none of the above situations is encountered, a positive reinforcement R_g is given, and the EGA is considered successful.

6 Computing Action Parameters

Let $\Gamma = \{\Pi_1, \Pi_2\}$ be a primitive pair for which the coplanar relationship is fulfilled. Let Γ_i be the i:th pair and $\mathbf{p}$ the plane defined by the coplanar relationship of the primitives of Γ_i. Let $\Lambda(\Pi)$ be the position of Π and $\Theta(\Pi)$ be the orientation of Π. The parameterization of the EGAs is given with the gripper normal $\mathbf{n}$ and the normal of the surface between the two fingers $\mathbf{a}$ as illustrated in Fig. 6. From this, the yaw, pitch and roll angles can be easily computed.

For EGA1, there will be two possible parameter sets given the primitive pair $\Gamma = \{\Pi_1, \Pi_2\}$. The parameterization is as follows:

$$\begin{aligned}
&\mathbf{p}_{\text{gripper}} = \Lambda(\Pi_i) \\
&\mathbf{n} = \nabla(\mathbf{p}) \\
&\mathbf{a} = \mathbf{perp}_{\mathbf{n}}(\Theta(\Pi_i)) / \parallel \mathbf{perp}_{\mathbf{n}}(\Theta(\Pi_i)) \parallel \quad \text{for } i = 1,2
\end{aligned}$$

where $\nabla(\mathbf{p})$ is the normal of the plane $\mathbf{p}$ and $\mathbf{perp}_{\mathbf{u}}(\mathbf{a})$ is the projection of $\mathbf{a}$ perpendicular to $\mathbf{u}$. That is $\mathbf{perp}_{\mathbf{u}}(\mathbf{a}) = \mathbf{a} - \mathbf{proj}_{\mathbf{u}}(\mathbf{a})$, where $\mathbf{proj}_{\mathbf{u}}(\mathbf{a})$ is defined according to (1).
For EGA2, there is only one parameter set.

$$\begin{aligned}
&\mathbf{d} = \Lambda(\Pi_2) - \Lambda(\Pi_1) \\
&\mathbf{p}_{\text{gripper}} = \Lambda(\Pi_1) + \mathbf{d}/2 \\
&\mathbf{n} = \nabla(\mathbf{p}) \\
&\mathbf{a} = \mathbf{n} \times \mathbf{d} / \parallel \mathbf{n} \times \mathbf{d} \parallel
\end{aligned}$$

For EGA3, there will be two possible parameter sets for $i = 1, j = 2$ and $i = 2, j = 1$.

$$\mathbf{d} = \Lambda(\Pi_j) - \Lambda(\Pi_i)$$
$$\mathbf{n} = \mathbf{d} / \parallel \mathbf{d} \parallel$$
$$\mathbf{p}_{\text{gripper}} = \Lambda(\Pi_i)$$
$$\mathbf{a} = \mathbf{n} \times \nabla(\mathbf{p})$$

For EGA4, there will be two possible parameter sets for $i = 1, j = 2$ and $i = 2, j = 1$. Where ε is a step size parameter that will depend on the gripper used.

$$\mathbf{d} = \Lambda(\Pi_j) - \Lambda(\Pi_i)$$
$$\mathbf{n} = \mathbf{d} / \parallel \mathbf{d} \parallel$$
$$\mathbf{p}_{\text{gripper}} = \Lambda(\Pi_i) - \nabla(\mathbf{p}) \cdot \varepsilon$$
$$\mathbf{a} = \mathbf{n} \times \nabla(\mathbf{p})$$

EGA5 will have the same parameters as EGA2 except that the gripper opening will be $\delta = \parallel \Lambda(\Pi_2) - \Lambda(\Pi_1) \parallel + \Delta$.

7 Limiting the Number of Actions

For a typical scene, the number of coplanar pairs of primitives is in the order of $10^3 - 10^4$. Given that each coplanar relationship gives rise to 8 different grasps from the five different categories, it is obvious that the number of suggested actions must be further constrained. Another problem is that coplanar structures occur frequently in natural scenes and only a small set of them suggest feasible actions, e.g. objects placed on a table create a lot of 3D line structures coplanar to the table but can not grasped directly by a grasping direction normal to the table. In addition, there exist many coplanar pairs of primitives affording similar grasps.

To overcome some of the above problems, we make use of the structural richness of the primitives. First, their embedding into collinear groups naturally clusters the grasping hypotheses into sets of redundant grasps from which only one needs to be tested. Furthermore, co–colority, gives an additional hypothesis for a potential grasp.

8 Using Grouping Information

From the 2D primitives (before stereo reconstruction) collinear neighbors can be found. The collinear neighbors can be mapped to corresponding 3D primitives. These small neighborhoods form the set of *small groups*, $\{g_1, g_2, ..., g_N\}$. The *large groups*, $\{G_1, G_2, ..., G_M\}$, are formed by the grouping of the small groups such that if Π_i and Π_j are part of group g_x and Π_j and Π_k is part of group g_y then g_y and g_x is part of the same large group G_z. Using this grouping information it is possible to add additional constraints on the generation of EGA s.

First, all primitives that are not part of a sufficiently large group G_i are discarded. Secondly, the relations co–planarity and co–colority between small groups of primitives are computed such that primitive $\Pi_i \in g_x$ and $\Pi_j \in g_y$ are only considered to have a co–planarity or co–colority relation if all primitives in g_x are coplanar or cocolor w.r.t all primitives in g_y. Finally, it is possible to constrain the generation of EGAs to only one EGA of each type for each large group.

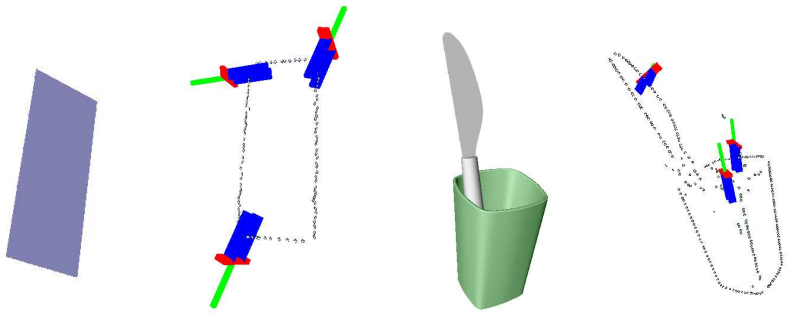


Fig. 7. Two example scenes designed for testing and a selection of the generated actions.

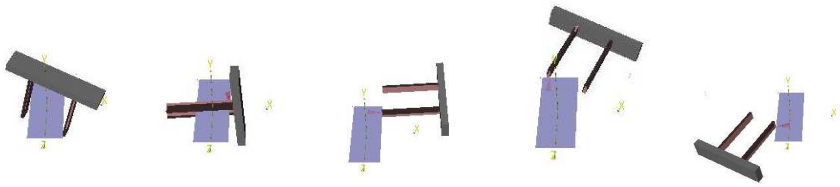


Fig. 8. Examples of tested grasps on a plate (from left): successful grasp using EGA5, and a few early failures using EGA1, EGA3 and EGA5, res5respectively.

9 Experimental Evaluation

Fig. 8, Fig. 9 and Fig. 10 show some of the grasps generated for the scenes evaluated here. Fig. 7 shows visual features generated by the stereo system and a selection of generated actions. Fig. 8 shows a simple plate structure for which the outer contour is generated since the object is homogeneous in texture. Fig. 9 shows a scene with a single, but a more complex object than the previous one. Fig. 10 shows two scenes with two (cup and knife) and three objects (box, cup and bottle9.

On each of the scene, after the spatial primitives have been extracted, elementary actions shown in Fig. 5 are tested. There are few reasons for which a certain grasp may fail:

- The system does not have the knowledge of whether the object is hollow or not, so testing EGA2 will results with a collision and thus failure.
- Since no surface is reconstructed, EGA1 will fail for hollow objects which are grasped from “below”.
- If the hand, during the approach, detects a collision on one of the fingers, the grasping process is stopped. In reality, this grasp may happen to be successful anyway if the object is moved so that it is centered between the fingers.

Table 1 summarizes the results for the generated success rate regarding a number of successful grasps given no knowledge of the object shape. We note that the results are a summary of an extensive experimental evaluation since, given different types and combinations of spatial primitives all generated actions had to be evaluated. It can be seen that for a scene of low

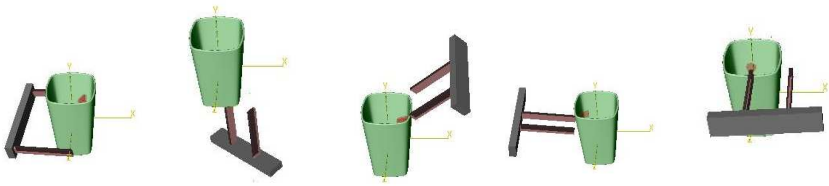


Fig. 9. Examples of tested grasps on a cup (from left): a successful grasp using EGA1, and a few early failures using EGA1, EGA1, EGA2 and EGA3, respectively.

Scene	gr	pl+gr	col+gr	gr+pl+col
Plane	90% (9/10)	67% (4/6)	86% (6/7)	80% (4/5)
Cup	21% (14/66)	27% (10/37)	27% (13/49)	29% (7/24)
Cup/Kn	20% (9/45)	9% (3/32)	26% (9/35)	15% (3/20)
3 objects	6%(27/434)	7% (7/98)	9% (12/139)	9% (9/53)

Table 1. Experimental evaluation of the grasp success rate where the following notation is used: pl (co-planarity), gr (grouping), cl (co-colority) and (succesfull/tested) grasps.

complexity (plate) the average number of successful grasps is close to 80%. For more complex scenes this number is dependant on the number and type of objects. It is also important to note not only the percentage but the number of evaluated grasps. Although, in some cases, the success rate is lower when primitives are integrated, there are much fewer hypotheses tested. These results should also be considered together with the results presented in Table 2 where we show how the integration of grouping, co-colority and co-planarity affects the number of generated hypotheses (affordances). Another thing to point out related to Table 1 is that most of the unsuccessful grasps happened due to an "early failure" such as that a contact was detected before the grasp was executed. Again, this failure may in some cases result with a successful grasp anyway. Another big source of failure was that there was nothing to lift, i.e. EGA3 could not have been applied.

Scene	(no gr)	(no gr)+pl	(no gr)+col	(no gr)+pl+co	gr	gr+pl	gr+col	gr+pl+coll
Plane	46 224	35 608	38 512	30 224	80	48	56	40
Cup	172 224	96 112	89 392	56 120	528	296	392	192
Cup/knife	269 360	140 920	139 136	79 104	360	256	280	160
3 objects	927 368	303 960	315 336	166 008	3472	784	1112	424

Table 2. The number of generated action hypotheses where the following notation is used: no gr (no grouping), pl (co-planarity), gr (grouping), cl (co-colority).

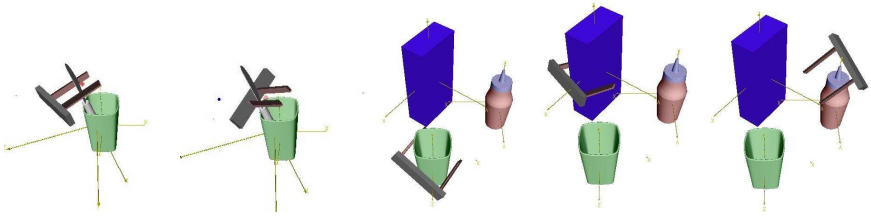


Fig. 10. Examples of successful grasps with two and three objects.

10 Conclusions

Robots should be able to extract more knowledge through their interaction with the environment. The basis for this interaction should not be a detailed model of the environment and lots of *a-priori* knowledge but the robot should be engaged in an exploration process through which it can generate more knowledge and more complex representations. In this paper, we have presented one of the building blocks necessary in such a system.

In particular, we have designed an early grasping system, based on a set of innate reflexes and knowledge about its embodiment. We relied on 3D information based on primitives extracted online and showed how the structural richness of primitives can be used for an efficient reduction of grasping hypotheses while keeping relevant ones. Rather than dealing with high quality grasps on a constrained set of known objects, we have demonstrated that the system is able of generating a certain percentage of successful grasps on arbitrary objects. This is important for our future research that will develop complex learning schemes aiming at more sophisticated grasping strategies and knowledge representation.

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

References

1. Azad, P., Asfour, T., Dillmann, R.: Combining appearance-based and model-based methods for real-time object recognition and 6d localization. In: IEEE International Conference on Intelligent Robots and Systems (2006)
2. Bicchi, A., Kumar, V.: Robotic grasping and contact: A review. In: IEEE International Conference on Robotics and Automation, pp. 348–353 (2000)
3. Ding, D., Liu, Y.H., Wang, S.: Computing 3-d optimal formclosure grasps. In: IEEE International Conference on Robotics and Automation, pp. 3573 – 3578 (2000)
4. Fitzpatrick, P., Metta, G., Natale, L., Rao, S., Sandini, G.: Learning About Objects Through Action - Initial Steps Towards Artificial Cognition. In: IEEE International Conference on Robotics and Automation, pp. 3140–3145 (2003)
5. Hauck, A., Rüttinger, J., Sorg, M., Färber, G.: Visual Determination of 3D Grasping Points on Unknown Objects with a Binocular Camera System. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 272–278 (1999)
6. http://www.barrett.com/robot/products/hand/handfram.htm:

7. Kamon, I., Flash, T., Edelman, S.: Learning Visually Guided Grasping: A Test Case in Sensorimotor Learning. IEEE Transactions on Systems, Man and Cybernetics **28**(3), 266–276 (1998)
8. Krüger, N., Felsberg, M.: An explicit and compact coding of geometric and structural information applied to stereo matching. Pattern Recognition Letters **25**(8), 849–863 (2004)
9. Krüger, N., Lappe, M., Wörgötter, F.: Biologically motivated multi-modal processing of visual primitives. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour **1**(5), 417–428 (2004)
10. Krüger, N., Wörgötter, F.: Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. International Symposium on Brain, Vision and Artificial Intelligence, Lecture Notes in Computer Science, Springer, LNCS 3704 pp. 157–166 (2005)
11. Miller, A.T., Allen, P.: Graspit!: A versatile simulator for grasping analysis. In: ASME International Mechanical Engineering Congress and Exposition (2000)
12. Miller, A.T., Knoop, S., P.K. Allen, H.I.C.: Automatic grasp planning using shape primitives. In: IEEE International Conference on Robotics and Automation, pp. 1824–1829 (2003)
13. Morales, A., Chinellato, E., Fagg, A.H., del Pobil, A.: Using experience for assessing grasp reliability. International Journal of Humanoid Robotics **1**(4), 671–691 (2004)
14. Morales, A., Recatalá, G., Sanz, P.J., del Pobil, Á.P.: Heuristic Vision-Based Computation of Planar Antipodal Grasps on Unknown Objects. In: IEEE International Conference on Robotics and Automation, pp. 583– 588 (2001)
15. Platt Jr, R., Fagg, A.H., Grupen, R.A.: Extending fingertip grasping to whole body grasping. In: International Conference on Robotics and Automation, pp. 2677 – 2682 (2003)
16. Pollard, N.S.: Parallel methods for synthesizing whole-hand grasps from generalized prototypes. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (1994)
17. Pollard, N.S.: Closure and quality equivalence for efficient synthesis of grasps from examples. International Journal of Robotic Research **23**(6), 595–613 (2004)
18. Pugeault, N., Wörgötter, F., Krüger, N.: Multi-modal scene reconstruction using perceptual grouping constraints. In: Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision, (in conjunction with IEEE CVPR 2006) (2006)
19. Rössler, B., Zhang, J., Knoll, A.: Visual Guided Grasping of Aggregates using Self-Valuing Learning. In: IEEE International Conference on Robotics and Automation, pp. 3912– 3917 (2002)
20. Rutishauser, M., Stricker, M.: Searching for Grasping Opportunities on Unmodeled 3D Objects. In: British Machine Vision Conference, pp. 277 – 286 (1995)
21. Stoytchev, A.: Behavior-Grounded Representation of Tool Affordances. In: IEEE International Conference on Robotics and Automation, pp. 3060–3065 (2005)

Robotics Group
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Technical Report no. 2008 – 2

A Hierarchical 3D Circle Detection Algorithm Applied in a Grasping Scenario

Emre Başeski, Dirk Kraft, Norbert Krüger

January 25, 2008

Title	A Hierarchical 3D Circle Detection Algorithm Applied in a Grasping Scenario
Author(s)	Emre Başeski, Dirk Kraft, Norbert Krüger
Publication History	

A Hierarchical 3D Circle Detection Algorithm Applied in a Grasping Scenario

Emre Başeski, Dirk Kraft, and Norbert Krüger

The Mærsk Mc-Kinney Møller Institute,
University of Southern Denmark,
Campusvej 55, DK-5230 Odense, Denmark
{emre, kraft, norbert}@mmmi.sdu.dk

Abstract. In this work, we approach the problem of 3D circle detection in a hierarchical representation which contains 2D and 3D information in the form of multi-modal primitives and their perceptual organizations. We use the information on different levels of the representation hierarchy in terms of semantic reasoning on higher levels leading to hypotheses that then become verified on lower levels by feeback mechanisms. The effects of uncertainties in visually extracted 3D information can be minimized by detecting a shape in 2D and calculating its dimensions and location in 3D. Therefore, we use the fact that the perspective projection of a circle on the image plane can be approximated as an ellipse and we create 3D circle hypotheses from 2D ellipses and the planes that they lie on. Afterwards, these hypotheses are verified in 2D, where the orientation and location information is more reliable than 3D. The algorithm is applied in a robotics application for grasping cylindrical objects.

1 Introduction

Circles are important structures in machine vision since they are a common feature for natural and human-made objects and they give more information than points and lines about the position and the pose of an object. In 3D vision, there are various ways of obtaining edge-like 3D entities (sparse stereo) from a stereo camera setup. Once the sparse stereo data is grouped with respect to a perceptual organization scheme, certain structures can be extracted from individual or combinations of these perceptual groups. Both, in dense and sparse stereo the correspondence finding phase in 3D reconstruction reduces the reliability of the information. Therefore, while detecting a certain structure like a 3D circle by using this kind of information, one needs to take into account the noise and uncertainty of the information.

The algorithms that are used to detect 3D circles can be grouped into three categories. The first category consists of *voting algorithms* like the Hough transform [1]. Due to the size of the parameter space, voting algorithms require much more memory and computational power than other algorithms. The second category contains *analytical algorithms* which use the geometric properties of circles (e.g., [2]). For laser-range data, this kind of algorithms run fast and are robust

because of the high-reliability of the information. Stereo vision on the other hand, introduces too many outliers and uncertainties that make the geometrical properties unstable. The last category involves the *fitting algorithms* that are traditionally based on minimizing a cost function which depends on a distance function that measures errors between given points and the fitted circle ([3–5]). The fitting process can be done either in 3D or in 2D. If it is done in 2D, the optimal plane for the given points is calculated and the points are projected onto that plane. If the fitting is done in 3D, the minimization starts with an initial estimate and tries to converge to the optimal circle. To guarantee convergence, a good initialization is required. This can be done by starting with multiple initializations, which decreases the computational efficiency drastically. One can reduce the parameter space as in [3] but the noisy nature of stereo vision data decreases the probability of convergence. Therefore, although fitting in 2D is a decoupled solution (plane fitting and curve fitting are handled separately), it is more advantageous in terms of efficiency and reliability for noisy data.

In this article, an algorithm which is based on fitting in 2D is presented. Note that, the common practice for such approaches is using only 3D information and its projection onto 2D. The main difference of our approach is, instead of using only 3D information, the representation hierarchy is used for different operations. Furthermore, there is a verification process, which is also done using different levels in the representation hierarchy.

In this work, the hierarchical representation presented in [6] is used. An example is presented in Fig. 1 which shows what kind of information exist in different levels of the representation. At the lowest level of the hierarchy, there is the image with its pixel values (Fig. 1(a)). At the second level, there exists the filtering results (Fig. 1(b)) which give rise to the multi-modal 2D primitives at the third level (Fig. 1(c)). At the third level, not only the 2D primitives but also 2D contours (Fig. 1(d)) are available that are created by using the perceptual organization scheme in [7]. The last level contains 3D primitives and 3D contours (Fig. 1(e-f)) created from 2D information of the input images.

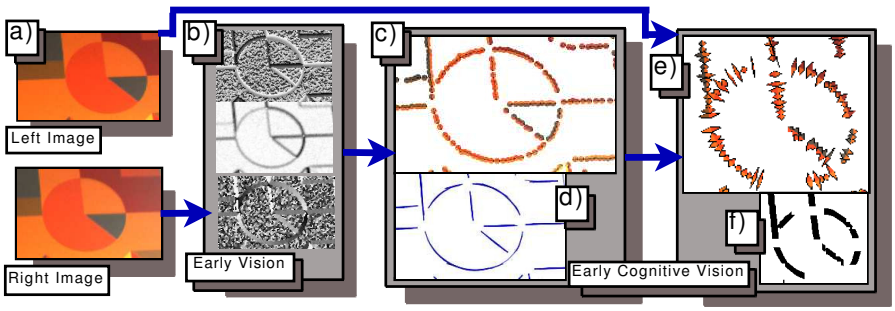


Fig. 1. Different type of information that is available in representation hierarchy **(a)** Original image **(b)** Filtering results **(c)** 2D primitives **(d)** 2D contours **(e)** 3D primitives **(f)** 3D contours

Since the reliability and the amount of data decreases as the level of representation hierarchy increases ([8]), lower levels should be used to verify the operations done in higher levels. For example, localization of a shape in 3D can be checked in 2D, once the perspective projection of the shape is known. Note that, there are more primitives and the orientation and location information is more reliable in 2D.

A part based object representation allows the association of actions to these parts. This leads to a transferability of actions from one object to another — if actions are learned from actual experience — and a possibility to define general actions for that specific part. Circles can be seen as one such part. We will be using them to define grasps that are working on cylindrical objects.

The rest of the article is organized as follows: In Sect. 2, the circle detection algorithm is introduced and some evaluation results on different scenarios are discussed. The experiments done on different objects in a grasping scenario where 3D dimension and location play an important role are presented in Sect. 3.

2 Circle Detection

The algorithm can be summarized in four steps as ellipse hypotheses creation, verification of these hypotheses, creating circles by transferring the verified hypotheses into 3D and verifying the created circles in 2D. The key idea is, 2D information is more reliable than 3D but we need 3D information to find orientation, radius and location of a circle. The verification is done in 2D so that the effect of the 3D information's low reliability can be minimized.

2.1 Computing Ellipse Hypotheses

Because of the correspondence problem in the 3D reconstruction process, the information in 2D can not be transferred to 3D completely. Therefore, contours in 2D contain more primitives than corresponding 3D contours and a 2D contour can contain projections of more than one 3D contour. These facts are the motivation to use 2D contours to search for 2D ellipses in the image. Another important fact is that, a single 2D contour may not be big enough to compute the ellipse that we are searching for. In Fig. 2(c) and (d), the ellipses fitted to contours in Fig. 2(b) are shown. Since the green contour is not big enough, the ellipse fitted to that contour is not the desired one.

Having too small data sets for fitting is a common problem originating from perceptual organization. To overcome this difficulty, a merging mechanism has been proposed in [9] which is based on proximity. Two curve segments are merged if the distance between their closest end points is smaller than a certain value (Fig. 2(e)). The first step of the algorithm starts with merging the 2D contours by using the proximity criterion. This merging operation creates a new set of 2D contours which contain the old 2D contours and their combinations.

Let $\mathcal{C}_i$ be the set of 3D contours whose projections on the image plane are contained in the 2D contour $\underline{c}_i$. Then, for the 3D contour $\underline{C}_i$, $P \cdot \underline{C}_i \in \underline{c}_j$ iff

Fig. 2. (a) Original image **(b)** Two contours on the circle (One is green and the other is white) **(c)** Fitted ellipse to the green contour in (b) **(d)** Fitted ellipse to the white contour in (b) **(e)** Two curves can be merged if min(d1,d2) is small enough

$\underline{C}_i \in \mathcal{C}_j$ (P is the projection matrix). Note that when two 2D contours are combined, it is represented as $\underline{c}_i + \underline{c}_j = \underline{c}_k^+$ and the set of 3D contours whose projections on the image plane are contained by the combination is represented as $\mathcal{C}_i + \mathcal{C}_j = \mathcal{C}_k^+$.

The ellipse hypotheses $\underline{e}_i$ that the 3D circles are based on are created from the combined contours where $\underline{c}_i^+$ is the 2D combined contour to which $\underline{e}_i$ is fitted. The ellipse fitting is done using the algorithm in [10] which is an ellipse specific least-squares fitting method. The fitted ellipses are represented using the general ellipse equation given in (1).

$$ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0 \tag{1}$$

2.2 Verification of Ellipse Hypotheses

Since we use the merged contours, the fitting procedure creates a lot of false ellipses as well as true ones. Therefore, not all the fitted ellipses are really in the scene. A true ellipse is shown in Fig. 3(c) which is fitted to the combination of the two red contours in Fig. 3(b) and a false ellipse is shown in Fig. 3(d) which is fitted to the combination of the bottom red and the green contour in Fig. 3(b).

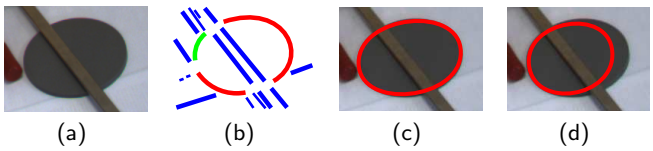


Fig. 3. (a) Input image **(b)**2D contours **(c)** A true ellipse **(d)**A false ellipse

The elimination of false ellipses is done by finding the *significance* [11] of the ellipses. The percentage of covered length of $\underline{e}_i$ is calculated from all 2D primitives (represented by π_j) that satisfy the following equations:

$$\|\pi_j - \underline{e}_i\| \leq \alpha_1 \tag{2}$$

$$|\arctan(\frac{d}{dx}e_{i|_{(\overline{x}_j,\overline{y}_j)}}) - \theta_j| \leq \alpha_2 \tag{3}$$

where α_1 and α_2 are thresholds, (2) is the distance between π_j and $\underline{e}_i$, (3) is the difference between the slope of $\underline{e}_i$ at $(\overline{x}_j, \overline{y}_j)$ and the orientation of π_j (represented by θ_j) and $(\overline{x}_j, \overline{y}_j)$ is the coordinate of the closest point on $\underline{e}_i$ to π_j. If π_j satisfies (2) and (3), its patch size (the diameter of the patch covered by the primitive) is added to the total covered length of $\underline{e}_i$. If the percentage of total covered length of $\underline{e}_i$ with respect to its perimeter is higher than a threshold, namely α_3, the ellipse is qualified as a true ellipse. The true ellipses for some scenes are shown in Fig. 4 where $\alpha_1 = 1\,pixel$, $\alpha_2 = 10°$ and $\alpha_3 = 60\%$.

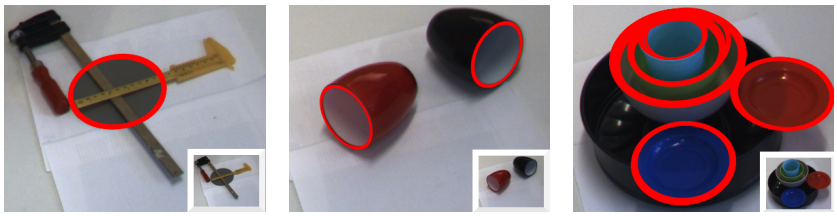


Fig. 4. Some true ellipse examples

2.3 Computing 3D Circle Hypotheses

Due to the fact that the perspective projection of a circle on the image plane can be approximated as an ellipse, it is possible to reconstruct the 3D circle, once the plane that the circle lies on is known. Therefore, at this point, to create 3D circles, the only further information we need is the plane p_i on which the circle that will be created from ellipse $\underline{e}_i$ lies. After calculating p_i, camera geometry can be used to find all the parameters of the 3D circle whose perspective projection is $\underline{e}_i$. Since we know the 2D contour $\underline{c}_i^+$ which gave rise to $\underline{e}_i$, it is possible to use the 3D contours $\mathcal{C}_i^+$ whose projections are contained by $\underline{c}_i^+$ to fit p_i. This operation gives the normal vector of the 3D circle as it is parallel to the normal vector of p_i. What is missing for the 3D circle is the center and the radius in 3D. For an ellipse represented as in (1), the center of the ellipse (x, y) is calculated as $(\frac{cd-bf}{b^2-ac}, \frac{af-bd}{b^2-ac})$.

Let (x_i, y_i) be the center of $\underline{e}_i$. Then, the intersection of p_i and the line passing from the camera center and $P^+[x_i\ y_i\ 1]^T$ gives the center of the 3D circle where P^+ is the pseudo-inverse of the projection matrix P. The procedure is illustrated in Fig. 5(a). We use the same methodology to calculate the radius of the circle. Take a random 2D primitive $\pi_j \in \underline{c}_i^+$. Let $[X_j\ Y_j\ Z_j]$ be the intersection of p_i and the line passing from the camera center and $P^+[x_j\ y_j\ 1]^T$. The distance between $[X_j\ Y_j\ Z_j]$ and the center of the circle gives the 3D radius. The 3D circles calculated in the this step can be represented in parametric form as:

$$R\cos(t)\boldsymbol{u} + R\sin(t)(\boldsymbol{n} \times \boldsymbol{u}) + \boldsymbol{c} \tag{4}$$

where $\boldsymbol{u}$ is a unit vector from the center of the circle to any point on the circumference; R is the radius; $\boldsymbol{n}$ is a unit vector perpendicular to the plane and $\boldsymbol{c}$ is the center of the circle.

Some results are presented in Fig. 5(b-c). Note that more than one combined contour can represent the same ellipse and they produce correct circles as well as some false ones because of the 3D reconstruction uncertainties. The false circles are eliminated in the next step.

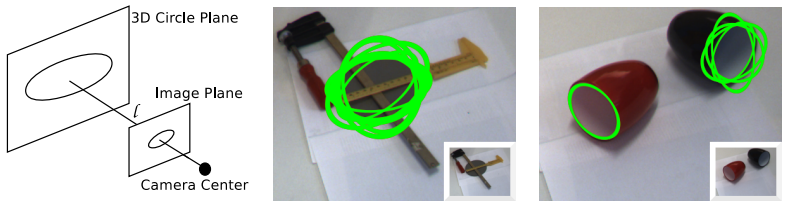


Fig. 5. (a) Calculation of the center of 3D circle **(b-c)**Projection of 3D circles on the image plane before verification

2.4 Final Selection of Circle Hypotheses

As the last step, our aim is to find which 3D circle is the best for ellipses that have been represented by more than one combined contour. Let $\mathcal{E}_i$ be the set of ellipses that are similar. It is impossible for them to have the same curve parameters so we can measure the similarity between two ellipses as a cost function depending on the distance between their centers, the difference of their perimeters and orientations. The main idea of the last step is to calculate the *significance* of ellipses which are projections of circles created from the ellipses in set $\mathcal{E}_i$. We do the evaluation in 2D since the amount and the reliability of data in this dimension is higher than 3D. To find the ellipse which is the perspective projection of a 3D circle, we can pick 5 points of the circle on the image plane and use the implicit equation of the conic through 5 points as in (5).

$$\begin{vmatrix} x^2 & xy & y^2 & x & y & 1 \\ x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ & & \cdots & & & \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \end{vmatrix} = 0 \tag{5}$$

The 5 points can be created from (4) for $t \in \{0, 80...320\}$. Equation 5 gives the generic equation of an ellipse as in (1). Therefore, we find the *significance* of these projected ellipses by using all 2D primitives π_j that satisfy (2) and (3). Some results are presented in Fig. 6

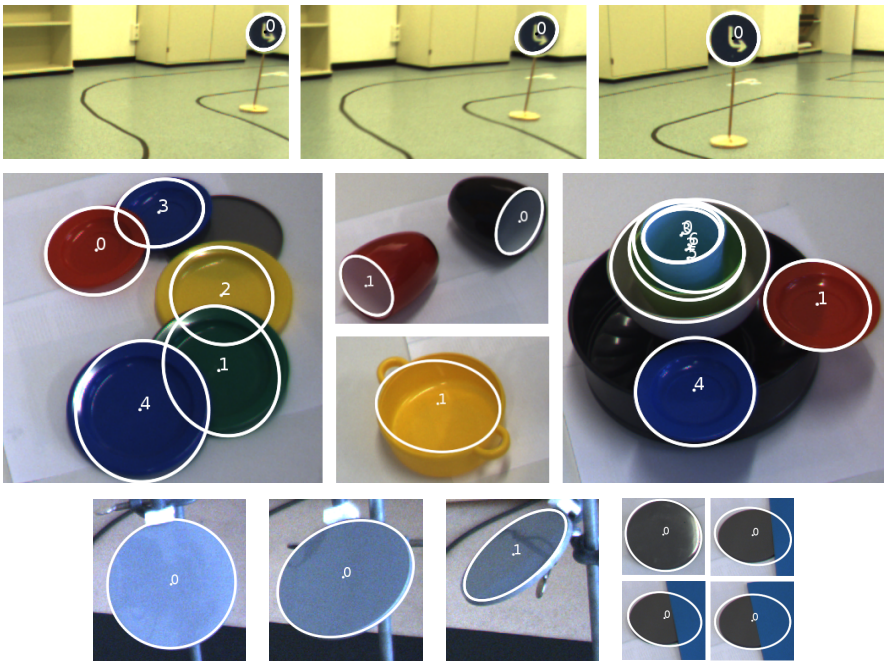


Fig. 6. 3D circle detection results on different scenarios. (White ellipses are the projections of 3D circles onto the image plane)

2.5 Problems

Although the algorithm is stable on tilted, partially covered and cluttered circles, perceptual organization can create problems in case of good continuation between circular and non-circular parts. Figure 7(b) illustrates a case, where the red 2D contour combines a circular and a non-circular part. In such cases, the remaining circular part (e.g., green contour in Fig. 7(b)) may create a valid ellipse hypothesis but transferring this hypothesis to 3D is heavily dependent on the plane that is fitted to the 3D points and usually this situation leads to incorrect 3D circles as shown in Fig. 7(c).

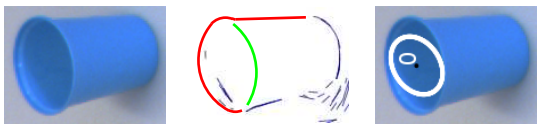


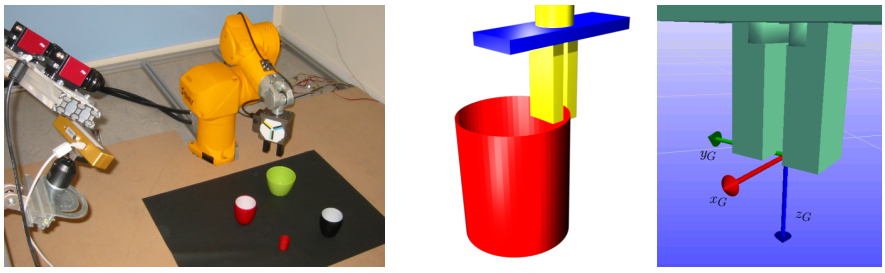
Fig. 7. **(a)**Original image **(b)** 2D contours corresponding to (a) **(c)** Detected 3D circle

3 Application in a Grasping Scenario

The algorithm described in the previous section is applied in a robot grasping application. In this section we describe the setup and use of this application to evaluate the circle detection.

3.1 System Description

The robotic system used consist of a six degree of freedom industrial robot (Stäubli RX-60B), a two finger parallel gripper (Schunk PG 70) and a Point Grey BumbleBee2 stereo camera (see Fig. 8(a)). The camera is calibrated relative to the robot coordinate system. Therefore the output of the above algorithm can be directly used for the computation of the grasping position.



(a) (b) (c)

Fig. 8. (a) Robot system consisting of six degree of freedom industrial robot, two finger gripper and two stereo camera systems (The lower camera systems was used for this work). (b) Grasp at the brim of the cylindrical object. (c) Gripper coordinate system.

3.2 Grasp Definition

For this work we selected one of the grasps defined in the grasping application to evaluate the quality of the circle detection. The cylindrical object is grasped on its brim (see Fig. 8(b)). The position of the grasp is expressed similar to the parametric form in (4). From this observation directly follows that there is actually not one possible grasp, but a one dimensional manifold of grasps (varying the grasp position around the circumference of the circle). Additionally the grasping depth h can be chosen according to the requirements of the scene. The position p of the grasper can therefore be defined as:

$$\boldsymbol{p} = R\cos(t)\boldsymbol{u} + R\sin(t)(\boldsymbol{n} \times \boldsymbol{u}) + \boldsymbol{c} - \boldsymbol{n}h \ . \tag{6}$$

Figure 8(c) shows the position and orientation of the grasper coordinate system defined at the end of the fingers. The grasper needs to be aligned in the following way: $\boldsymbol{z_G} = -\boldsymbol{n}$ and $\boldsymbol{y_G} = \cos(t)\boldsymbol{u} + \sin(t)(\boldsymbol{n} \times \boldsymbol{u})$. While the gripper opening can be defined as $d = \max(2R, d_{max})$.

3.3 Evaluation

Figure 9 shows a number of scenarios where the gripper is moved to the grasping position computed based on the circle information ($h = 2\,cm$, t was used in a standard configuration except when this would have lead to a collision). The different setups show that our system is able to cope with different levels of complexity.

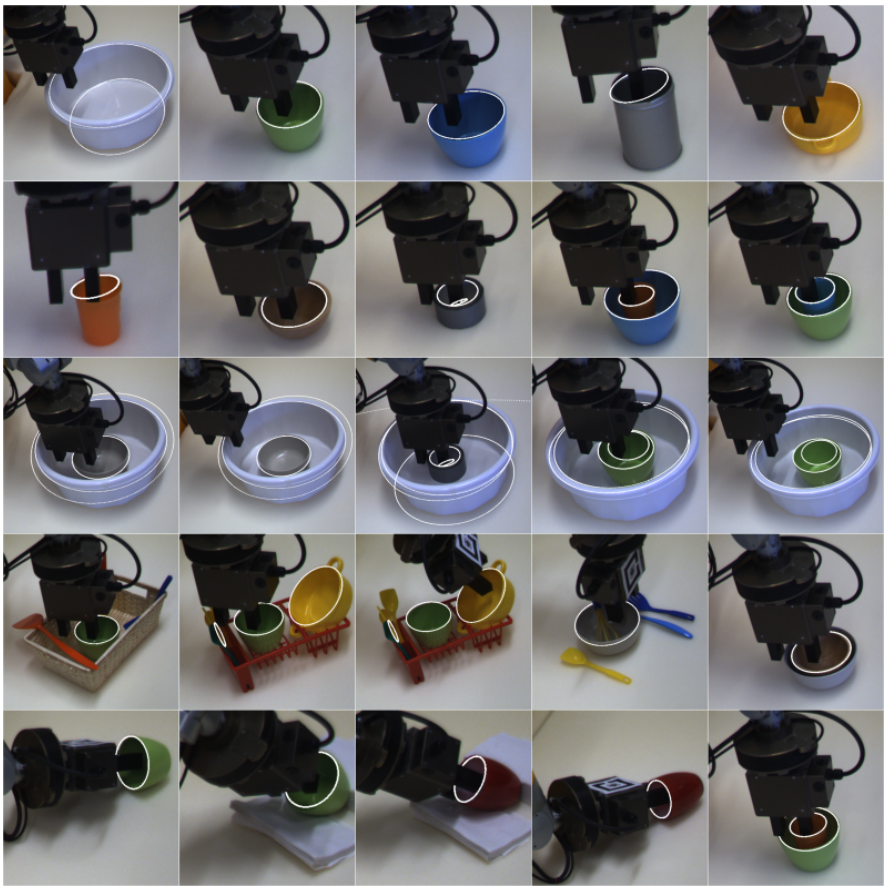


Fig. 9. Detected circles and applied grasps. The circles were drawn into the images and the occluded parts were corrected afterward to improve the readers scene understanding. The scenes are of different complexity, starting out with single objects, going to objects included in each other, multiple (and more complex) objects and finally tilted single objects.

4 Conclusion

We have discussed a 3D circle detection algorithm which makes use of different aspects of 2D and 3D information for hypothesis generation and verification. To be able to cope with the uncertainties of sparse stereo data, 3D circles are localized in 3D by considering 2D hypotheses and verified in 2D, where the information is more reliable. The potential of the approach has been shown on a grasping application for different scenarios. As a future work, the problem of combining circular and non-circular parts will be handled by splitting 2D contours with respect to junctions and 3D structure of the contour.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)
2. Xavier, J., Pacheco, M., Castro, D., Ruano, A., Nunes, U.: Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. (2005) 3930–3935
3. Jiang, X., Cheng, D.C.: Fitting of 3D Circles and Ellipses Using a Parameter Decomposition Approach. In: 3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, IEEE Computer Society (2005) 103–109
4. Chernov, N., Lesort, C.: Least Squares Fitting of Circles. J. Math. Imaging Vis. **23**(3) (2005) 239–252
5. Shakarji, C.: Least-Squares Fitting Algorithms of the NIST Algorithm Testing System. Res. Nat. Inst. Stand. Techn. **103** (1998) 633–641
6. Krüger, N., Lappe, M., Wörgötter, F.: Biologically Motivated Multi-modal Processing of Visual Primitives. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour **1**(5) (2004) 417–428
7. Pugeault, N., Wörgötter, F., Krüger, N.: Multi-modal Scene Reconstruction Using Perceptual Grouping Constraints. In: Proceedings of the IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06). (2006)
8. Pugeault, N., Kalkan, S., Başeski, E., Wörgötter, F., Krüger, N.: Reconstruction Uncertainty and 3D Relations. In: Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08). (2008)
9. Ji, Q., Haralick, R.M.: A Statistically Efficient Method for Ellipse Detection. In: ICIP (2). (1999) 730–734
10. Pilu, M., Fitzgibbon, A., Fisher, R.: Ellipse-Specific Direct Least-Square Fitting. In: In Proc. IEEE ICIP. (1996)
11. Lowe, D.G.: Three-Dimensional Object Recognition from Single Two-Dimensional Images. Artificial Intelligence **31**(3) (1987) 355–395

Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping

Kai Huebner, Steffen Ruthotto and Danica Kragic

***Abstract*— Thinking about intelligent robots involves consideration of how such systems can be enabled to perceive, interpret and act in arbitrary and dynamic environments. While sensor perception and model interpretation focus on the robot's internal representation of the world rather passively, robot grasping capabilities are needed to actively execute tasks, modify scenarios and thereby reach versatile goals. These capabilities should also include the generation of stable grasps to safely handle even objects unknown to the robot. We believe that the key to this ability is not to select a good grasp depending on the identification of an object (e.g. as a cup), but on its shape (e.g. as a composition of shape primitives). In this paper, we envelop given 3D data points into primitive box shapes by a fit-and-split algorithm that is based on an efficient Minimum Volume Bounding Box implementation. Though box shapes are not able to approximate arbitrary data in a precise manner, they give efficient clues for planning grasps on arbitrary objects. We present the algorithm and experiments using the 3D grasping simulator *GraspIt!* [1].**

I. INTRODUCTION

In the service robot domain, researchers and programmers provide each robot with manifold tasks to do in order to aid and support, e.g. clearing a table or fill a dishwasher after lunch. The knowledge about such aims might be either hard-coded or learned in a more intelligent manner, e.g. by a person teaching the robot how to clear a table. Such scenarios are known as Learning- or Programming-by-Demonstration applications. However, whether in an office, in health care or in a domestic scenario, a robot has to finally operate independently to satisfy various claims. Thus, the handling of objects is a central issue of many service robot systems. Robot grasping capabilites are therefore essential to actively execute tasks, modify scenarios and thereby reach versatile goals in an autonomous manner.

For grasping, numerous approaches and concepts have been developed over the last decades. Designing grasping systems and planning grasps is difficult due to the large search space resulting from all possible hand configurations, grasp types, and object properties that occur in regular environments. Early work on contact-level grasp synthesis focused mainly on finding a fixed number of contact locations without regarding hand geometry [2]. Considering specifically object manipulation tasks, the work on automatic grasp synthesis and planning is of significant relevance [3], [4], [5]. The main issue here is the automatic generation of stable grasps assuming that the model of the hand is known and that certain assumptions about the object (e.g. shape, pose) can be made. 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 [4]. It is obvious that knowledge about the object shape, as also the task on hand, is quite meaningful for grasp planning [6].

This is important for our scenario, in which we aim at providing a robot actuator system with a set of primitive actions, like *pick-up*, *push* or *erect* an arbitrary object on a table. For performing such basic actions, an object has to be modeled from 3D sensory input, e.g. from range or dense stereo data. However, we state the question up to which detail this is necessary in terms of grasping.

II. MOTIVATION

Modeling range data is a crucial, but also difficult task for robot manipulation. The source data offered by range sensors or dense stereo camera systems is a more or less distorted and scattered cloud of 3D points of the scenario. A higher-level representation of these points as a set of shape primitives (e.g. planes, spheres or cylinders) obviously gives more valuable clues for object recognition and grasping by compressing information to their core. Most approaches that consider this problem are likewise bottom-up, starting from point-clouds and synthesizing object shapes by using superquadrics (SQs). Superquadrics are parametrizable models that offer a large variety of different shapes. Considering the problem of 3D volume approximation, only superellipsoids are used out of the group of SQs, as only these represent closed shapes. There is a multitude of state-of-the-art approaches based on parametrized superellipsoids for modeling 3D range data with shape primitives [7], [8], [9], [10].

Assuming that an arbitrary point cloud has to be approximated, one SQ is not enough for most objects, e.g. a screw or an office chair (see Fig. 1). The more complex the shape is, the more SQs have to be used to conveniently represent its different parts. However, good generality is not possible with few parameters for such cases [7]. Besides the advantages of immense parametrization capabilities with at least 11 parameters, intensive research on SQs has also yielded disadvantages in two common strategies for shape approximation. The first strategy is region-growing, starting with a set of hypotheses, the *seeds*, and let these adapt to the point set. However, this approach has not proved to be effective [8] and suffers from the refinement problem of the seeds [10]. The second strategy uses a split-and-merge

All authors are or were with the Computer Vision & Active Perception Lab., Computer Science and Communication, Royal Institute of Technology (KTH), 100 44 Stockholm, Sweden {khubner, ruthotto, danik}@kth.se

technique. Splitting up a shape and merging parts again is more adapted to unorganized and irregular data [8].

Independent of the strategy used, the models and seeds, respectively, have to be fitted to the 3D data. This is usually done by least square minimization of an inside-outside fitting function, as there is no analytical method to compute the distance between a point and a superquadric [9]. Thus, SQs are though a good trade-off between flexibility and computational simplicity, but sensitive to noise and outliers that will cause imperfect approximations. This is an important issue, as our work is oriented towards the use of dense stereo accompanied by highly distorted and incomplete data.

We observed that modeling 3D data by shape primitives is a valuable step for object representation [11]. Sets of such primitives can be used to describe instances of the same object classes, e.g. cups or tables. However, it is not our aim to focus on such high-level classifications or identification of objects, but on grasping. We moreover approach a deeper understanding of objects by interaction instead of observation for that purpose, e.g., if there is an object that can be picked up, pushed and filled, it can be used as a cup. Processing an enormous number of data points takes time, both in approaches that use the raw points for grasp hypotheses and in those that approximate as good as possible by shape primitives. In this context, a question remains: *how rudimentary can a model of a thing be in order to be handled successfully and efficiently?* While comparable work is placed mostly at extrema of this scale, e.g. by using pairs of primitive feature points [12] or a-priori known models for each object [11], we are interested in looking into which primitive shape representations might be sufficient for the task of grasping arbitrary, unseen objects.

We believe that a mid-level solution is a promising trade-off between good approximation and efficiency for this purpose. Complex shapes are difficult to process, while the simple produce worse approximation. However, we can access valuable methods to handle approximation inaccuracies for grasping like haptic feedback, visual servoing and advanced grasp controllers for online correction of grasps. We prefer general fast online techniques instead of pre-learned offline examples, thus the algorithm's efficiency is the more important issue. Unknown objects are hardly parametrizable but need real-time application for robot grasping. A computation in terms of minutes for a superquadric approximation is therefore not feasible.

We adopt these motivations to propose an algorithm based on boxes as a mid-level representation. In our approach, we combine different incentives on simplicity of boxes, efficiency of hierarchies and fit-and-split algorithms:

1) We aim for *simplicity* stating the question if humans approach an apple for grasping with their hand in another way as they approach a cup, or a pen in another way as a fork? While there are surely differences in fine grasping and task dependencies, differences in approaching these objects seem quite marginal.
2) Computational efficiency of *hierarchies* was pointed out in several other approaches that compose models with use of superquadric primitives [7], [9], [13].
3) While seed growing as a bottom-up strategy has several drawbacks, and a split-and-merge strategy both needs top-down (split) and bottom-up (merge), *fit-and-split* algorithms are purely top-down and thereby iteratively implementable in a one-way hierarchical manner.

Following our primary incentive, we chose boxes as a very simple and roughly approximating representation.

III. ALGORITHM

A. Computing Bounding Boxes

The algorithm of minimum volume bounding box computation proposed by Barequet and Har-Peled [14] will form the base for our approach. Given a set of n 3D points, the implementation of the algorithm computes their Minimum Volume Bounding Box (MVBB) in $O(n \log n + n/\varepsilon^3)$ time, where ε is a factor of approximation. The algorithm is quite efficient and parametrizable by several optimizations. Performing the computation on an arbitrary point cloud, a tight-fitting, oriented MVBB enclosing the data points is produced (see the example in Fig. 1).

B. Decomposition of MVBBs

Based on this algorithm, we aim at iteratively splitting the box and the data points, respectively, such that new point sets yield better box approximations of the shape. Iterative splitting of a root box corresponds to the build-up of a hierarchy of boxes. Gottschalk *et al*. [15] present the OBBTree (Oriented Bounding Box Tree) for this purpose, where the goal is efficiently collision detection between polygonal objects. The realization of the splitting step is quite straightforward: each box is cut at the mean point of the vertices, perpendicular to the longest axis. This is done iteratively, until a box is not dividable any more. Similar work on division of polygonal structures for grasping has been proposed by others [16], [17]. In our case, these strategies are suboptimal or less applicable. Splitting into many small boxes is against our aim of approximating a shape with as few boxes as possible. Additionally, though the MVBB algorithm is efficient, a fitting step after each splitting consumes valuable computation time. Finally, in our application both the splitting at the mean point is not optimal and we can not access polygonal structures, but point clouds only. Thus, another heuristic to find a "good" split is needed.

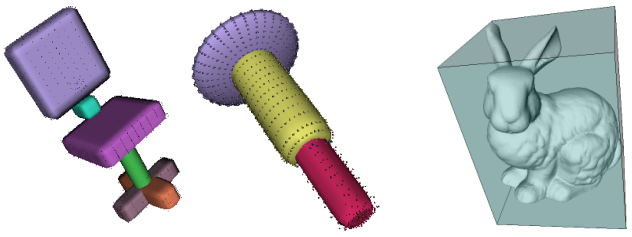


Fig. 1. Left: Examples of range data approximated by sets of superquadrics [8]. Right: The Stanford bunny model and the root MVBB of its vertices.

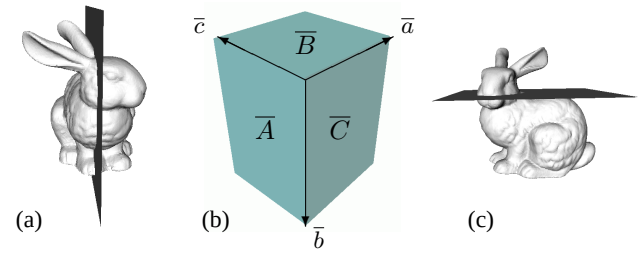


Fig. 2. (a) A mean cut of the bunny model. (b) We restrict to box parallel cutting planes. (c) A good cut parallel to the root MVBB plane $\overline{B}$.

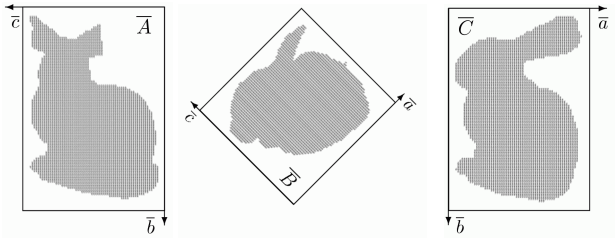


Fig. 3. Bunny sample projections onto the three faces of the root box (Fig. 1) according to the face-parallel cutting scheme in Fig. 2(b).

Therefore, we will have to define what a "good" split is. Fig. 2(a) shows a mean cut, which obviously is not good for our task. It does not improve the approximation with boxes of both new halfs, but is also not intuitive in terms of dividing the bunny in semantic parts, e.g. head and body. Even with planar cuts, finding the best intuitive one would correspond to an extensive search and comparison of a lot of planes, differing in position and orientation. Therefore, we decide to test only the planes parallel to the parent MVBB, like Fig. 2(b) and (c) show. As a measure of a good split, we can consult the relation of the box volume before and after the splitting: a split of the parent box is the better, the less volume the two resulting child MVBBs will include. This is intuitively plausible, as shape approximation is better with highly tight-fitting boxes.

C. Computing the Best Split

As motivated, we just test planes parallel to the three box surfaces for the best splitting plane. Each MVBB has six sides, whereof opposing pairs are parallel and symmetric. Inbetween each of these pairs, we can shift a cutting plane. Fig. 2(b) depicts this restriction on a splitting parallel to $\overline{A}$, shifted by a distance a, and $\overline{B}$ by b and $\overline{C}$ by c, respectively. A computation of new MVBBs for each value of the *split parameters* a, b and c would take a lot of computational effort. Therefore, we estimate the best cut by first projecting the data on 2D grids which correspond to the surfaces $\overline{A}$, $\overline{B}$ and $\overline{C}$. The bunny sample data projections onto the three surface grids of the root MVBB are shown in Fig. 3, reducing the problem of splitting a 3D box by a surface-parallel plane to splitting a 2D box by an edge-parallel line. For the sake of efficiency, it is thereby abstracted from the real 3D volume of the shape. The figure shows that there are six valid split directions left, two for each of the surfaces $\overline{A}$, $\overline{B}$ and $\overline{C}$.

As mentioned above, we define the best split as the one that minimizes the summed volume of the two partitions. Thus, we now test each discretized grid split along the six axes, using the split parameters. We define a split measure $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ with $\overline{\mathcal{F}} \in \{\overline{A}, \overline{B}, \overline{C}\}$ being the projection plane to split, $\overline{f}$ being one of the two axes that span $\overline{\mathcal{F}}$, and i as the grid value on this axis that defines the current split. Consequently, we have six possible split measures

$$\begin{aligned}
&\theta_1(\overline{A}, \overline{c}, i_1),\ i_1 \in \mathbb{N}^{<c_{\max}}, \theta_2(\overline{A}, \overline{b}, i_2),\ i_2 \in \mathbb{N}^{<b_{\max}},\\
&\theta_3(\overline{C}, \overline{a}, i_3),\ i_3 \in \mathbb{N}^{<a_{\max}}, \theta_4(\overline{C}, \overline{b}, i_4),\ i_4 \in \mathbb{N}^{<b_{\max}},\\
&\theta_5(\overline{B}, \overline{a}, i_5),\ i_5 \in \mathbb{N}^{<a_{\max}}, \theta_6(\overline{B}, \overline{c}, i_6),\ i_6 \in \mathbb{N}^{<c_{\max}} \qquad (1)
\end{aligned}$$

to compare. Their minimum gives reason to the best split. The minimization of each $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ is implemented as follows. For each i that cuts $\overline{\mathcal{F}}$ perpendicular to $\overline{f}$ in two rectangular shapes, we compute the two resulting minimum grid areas by lower and upper bounds. The i that yields the minimum value is the best cut of $\overline{\mathcal{F}}$ along $\overline{f}$. $\theta(\overline{\mathcal{F}}, \overline{f}, i)$ is computed as the fraction between the whole projection rectangular and the sum of the two best cut rectangles. Though this is a very approximative method, it is quite fast, as rectangle volume and bounds are easy to generate. The best bunny cuts for which rectangular volume and the corresponding values $\theta_{1\ldots6}$ are minimal are shown in Fig. 4.

D. Building a Fit-and-Split Hierarchy

According to the best split θ^*, which would be θ_1 or θ_2 in this exemplary case, the original point cloud can be divided into two subsets of the data points. These can be used as inputs to the MVBB algorithm to produce two child MVBBs of the root MVBB. In this way, the complete fit-and-split method can iteratively be performed. It is important to note that by MVBB re-computation, the MVBBs will greatly differ in orientation and scale from the box cuts in Fig. 4.

Additionally, the previous step of cutting along one of the six directions is just equal to computing an approximative gain value, for the purpose of efficiency. As an iteration breaking criterion, we now subsequently test the real MVBB volume gain Θ^* of the resulting best split measure θ^*. Therefore, we compute the gain in volume defining

$$\Theta^* = \frac{V(\mathbf{C}_1) + V(\mathbf{C}_2) + V(\mathbf{A}^{\setminus\mathbf{P}})}{V(\mathbf{P}) + V(\mathbf{A}^{\setminus\mathbf{P}})}, \qquad (2)$$

where $\mathbf{A}$ is the complete set of boxes in the current hierarchy, $\mathbf{P}$ is the current (parent) box, $\mathbf{C}_1$, $\mathbf{C}_2$ are the two child boxes produced by the split, and V being a volume function.

We decide further process on two constraints. First, if the gain is too low, a split is not valuable. For this purpose, we include a threshold value t. The precision of the whole approximation can be parametrized by simply preventing a split if Θ^* exceeds t. Second, we do not preserve boxes in the hierarchy that include a very low number of points. By this process, noise in the point data can be handled.

It might also be important in this context that in Fig. 4, θ_6 would intuitively be a probably valuable next cut below the bunny's ear. However, the best split computation presented (Section III-C) will not find this cut. Finding this cut is not that simple, especially when distorted, sparse and insecure

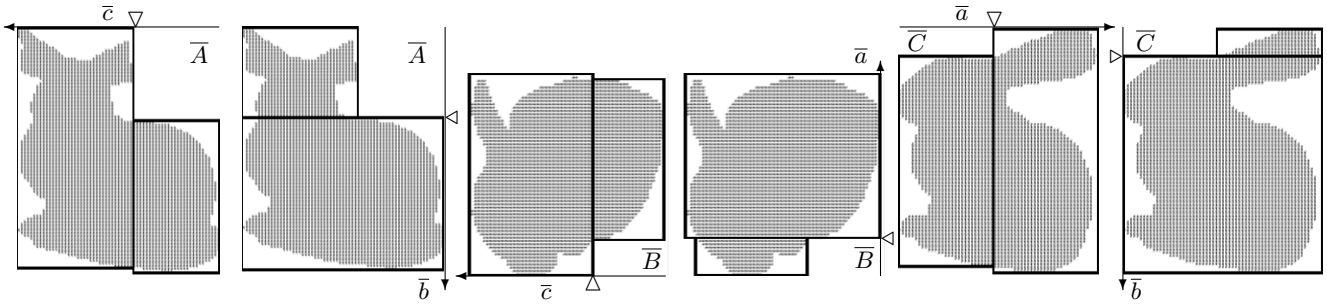


Fig. 4. Best cuts along the six box directions and cut positions i marked by triangles. The corresponding volume values $\theta(\overline{\mathcal{F}},\overline{f},i)$ are presented below.

data is provided. An add-on for the solution of this problem would therefore be more complex and time-consuming. The bunny is a very ideal model, as it is artificial, complete, and data points are very dense. As it is our aim to evaluate our algorithm also on real sensory data, we can not assume such ideal conditions and do not handle such situations presently.

IV. EXPERIMENTS

A. MVBB Splitting Evaluation

In the following, we present some experiments for the proposed fit-and-split algorithm. For all experiments, we fix the two original MVBB approximation parameters (see [14]). The grid parameter defines orientations that induce bounding box approximation in an exhaustive way, so we keep it small at 3. We decide to sample sets of 200 points, so even very large point clouds are reduced and efficiently handled. We found that these settings provide a good trade-off between quality and efficiency of each split for our application. The main parameter that we are going to change in the experiments is the gain threshold t.

We evaluate the behaviour of the algorithm on several types of input data by taking both ideal point clouds emerging from complete and unnoisy simulative vertice models (4 models) and real laser scan excerpts (5 scans) as input data. The latter is therefore incomplete and noisy, but at least regular due to the scan sampling. One sample is produced from a stereo vision system that offers three-dimensional points by disparity, including incomplete, noisy and irregular data. Fig. 5 shows these samples divided with different gain thresholds $t \in \{0.90, 0.94, 0.98\}$. The corresponding overview on point sets, computation time and number of boxes for the groups is given in Tab. I.

B. MVBB Grasping Evaluation

The best way to find a good grasp is said to be grasp candidate simulation [4], [9]. Miller *et al*. have simulated pre-models and shape primitives using their public grasp simulation environment *GraspIt!* [4]. So we also base our evaluation on model-based grasping in *GraspIt!*.

The first iteration, performed as proposed in Section III-D, yields the root node of the box tree. The root box has six faces, each of which we use for four grasp hypotheses parallel to its spanning edges. For symmetric grippers these could be reduced to two grasp hypotheses, but as we will use an asymmetric 5-finger hand model [18] in our simulation, we take these four. After the grasps on the root box have been performed, we apply the decomposition algorithm to produce MVBB approximations with gain parameters 0.90, 0.94 and 0.98 for the pure model data, Fig. 5(a)-(c), only. All faces are then collected from a final approximation, before occluded and ungraspable ones are removed. The applied grasping method is simple here: each initial position is set to a constant distance from the face's center aligned to its normal, i.e. the approach vector is the negative face normal. The hand is set to an intuitively good pose to have a large opening angle towards the object. We let the hand approach along the normal until a contact is detected. After contact, the hand retreats a small distance before we call GraspIt!'s auto-grasp function which uniformly closes the fingers of the hand. When all fingers are in contact with the object, we evaluate the two standard grasp quality measures that come with GraspIt!: ε, a worst-case epsilon measure for force-closure grasps, and V, an average case volume measure [1].

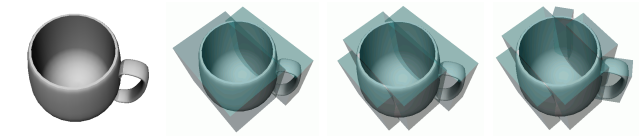
To compare the grasps that we get from this sequence, we compute a random "spherical" grasp evaluation for each model. Initial hand positions are placed on a sphere, with the approach vector oriented towards the object's center of mass and two spherical coordinates and a hand orientation angle configuring the hand's pose (discretized by steps of 10 deg).

We find that geometrical detection of blocked faces reduces the number of graspable faces drastically. Each spherical evaluation includes 22104 grasps. Referring to the grasp quality comparison between spherical and box evaluation for our models (Fig. 6), a_1 resulted from a test of only f=6 valid face grasps from the t =0.94 decomposition. Same pairs for

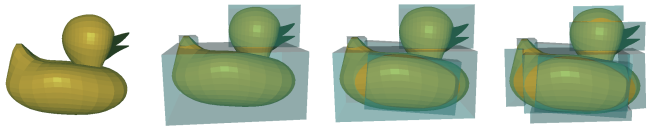
TABLE I

STATISTICS OF THE EXPERIMENTS PRESENTED IN FIG. 5.

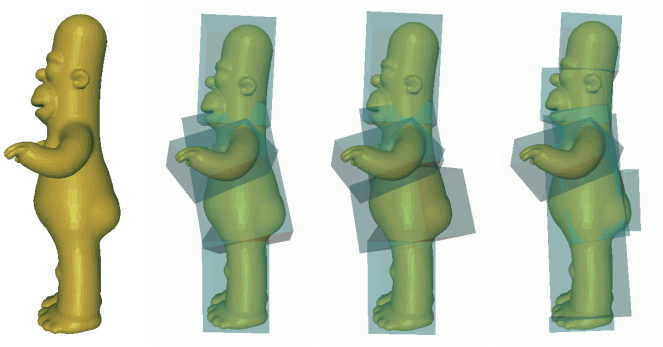
Model	#points	#boxes—sec (t=0.90)	#boxes—sec (t=0.94)	#boxes—sec (t=0.98)
Mug	1725	2—4	3—7	5—11
Duck	1824	3—7	5—9	9—14
Homer	5103	4—10	5—13	7—16
Bunny	35947	2—5	4—11	11—30
Stapler	313	2—2	2—2	2—2
Puncher	449	3—3	3—3	4—3
Can	1266	2—4	5—8	9—10
Phone	1461	3—5	4—5	9—12
Laptop	4199	3—7	4—8	6—15
Can2	9039	2—7	7—20	16—46



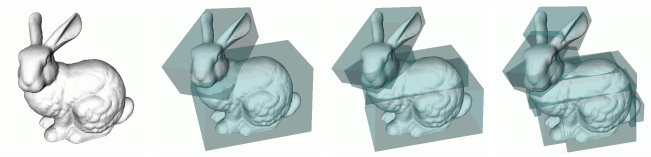
(a) Mug (model): MVBBs (2,3,5) produced with t=0.90, 0.94, 0.98.



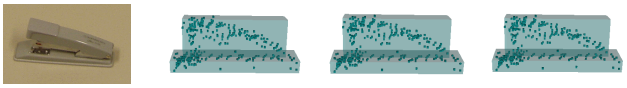
(b) Duck (model): MVBBs (3,5,9) produced with t=0.90, 0.94, 0.98.



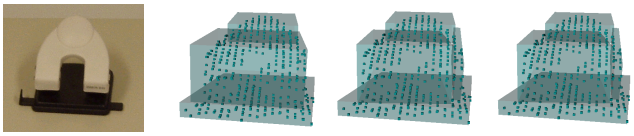
(c) Homer (model): MVBBs (4,5,7) produced with t=0.90, 0.94, 0.98.



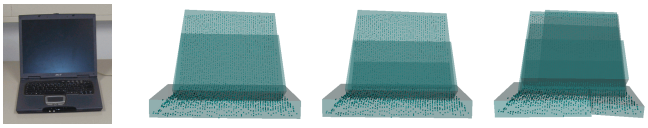
(d) Bunny (model): MVBBs (2,4,11) produced with t=0.90, 0.94, 0.98.



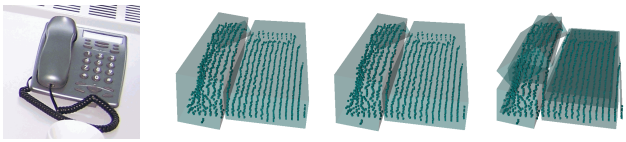
(e) Stapler (scan): MVBBs (2,2,2) produced with t=0.90, 0.94, 0.98.



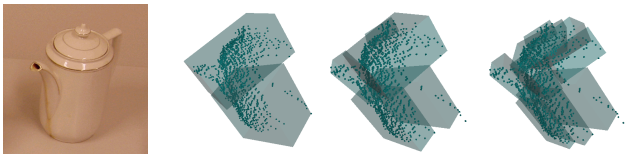
(f) Puncher (scan): MVBBs (3,4,4) produced with t=0.90, 0.94, 0.98.



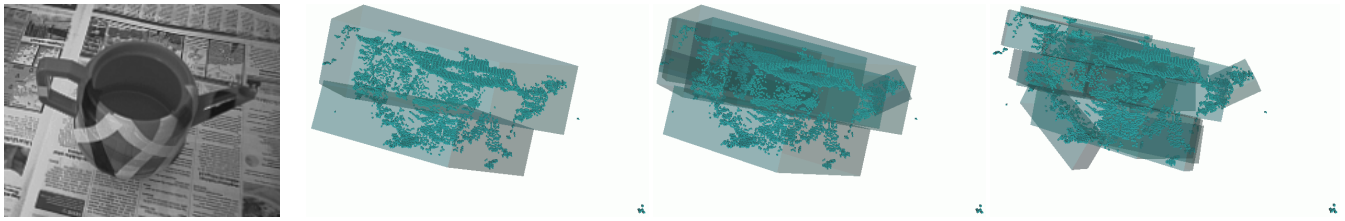
(g) Notebook (scan): MVBBs (3,4,6) produced with t=0.90, 0.94, 0.98.



(h) Phone (scan): MVBBs (3,3,9) produced with t=0.90, 0.94, 0.98.



(i) Can (scan): MVBBs (2,5,9) produced with t=0.90, 0.94, 0.98.



(j) Can2 (stereo): MVBBs (2,7,16) produced with t=0.90, 0.94, 0.98.

Fig. 5. Examples of box decomposition using different gain thresholds t=0.90, 0.94, 0.98, where numbers in brackets correspond to numbers of boxes. (a)-(d) are complete, dense and unnoisy 3D models. (e)-(i) result from incomplete, dense and less noisy, but manually pre-segmented range scans. (j) is produced by an incomplete, sparse and noisy, automatically pre-segmented stereo disparity point cloud.

the other depicted samples are: b_1=(16,Root), b_2=(32,0.94), c_1=(48,0.98) and c_2=(22,0.90). Concluding, the box decomposition effectively produces very few hypotheses which still feature good grasp quality. Note that only force-closure grasps ($\varepsilon > 0$) are drawn in Fig. 6.

V. DISCUSSION AND CONCLUSION

In our approach, we combined motivations known from the shape approximation and grasping literature. We prune the search space of possible approximations by rating and decomposing bounding boxes. Related work uses more complex superquadrics as approximation elements and confirm that grasp planning on finer components is likely to find better grasps than returning the first stable grasp [9]. This intuitively corresponds to the "grasping-by-parts" strategy. This strategy also underlies the presented approach of MVBB decomposition. In this paper, we proposed MVBBs as an efficient and valuable box decomposition on a fit-and-split strategy. As the presented approach is hierarchical, it is also possible to use dependencies between boxes and granularities of different hierarchical levels for shape approximation. Thus, the processing of shape approximation can be controlled and run parallel to the execution of a grasp.

The trade-off of our approach is higher efficiency and simplicity for the price of precise shape approximation. However, we claim that exact approximation may not be necessary for grasping tasks. A wider evaluation of this claim will be one of our next steps. Our approach is therefore

grounded on box representation and decomposition with an efficient splitting criterion. The resulting box representation offers fast computational techniques for common problems, e.g. collision detection, neighborhood relations, etc., valuable for efficient further analysis. This analysis will become important for a next step towards grasping objects. Managing valid grasps will not only be dependent on the box faces, but also on the whole constellation of boxes.

Another issue in this context will be task dependency. A grasp might depend on different types of tasks, e.g. to pick up a cup and place it somewhere else might yield a different grasping action as to pick it up to show it or hand it over to someone. Such grasp semantics might be mapped to boxes in the set, e.g. "grasp the *biggest box* for a good grasp to stably move the object", "grasp the *smallest box* for a good grasp to show a most unoccluded object to a viewer / a camera" or "grasp the *outermost box* for a good grasp to hand over to another human / another robot", where the latter are said to be quite valuable for applications that are based upon interacting with objects before the exploration and recognition stage. Future work will focus on how the presented box representation provides a good and easy-to-use interface to such applications.

VI. ACKNOWLEDGMENTS

This work was supported by EU through the project PACO-PLUS, IST-FP6-IP-027657. The authors would like to thank Michael Wünstel (Universität Bremen) for providing the raw partial scan data (Fig. 5(e)-(i)).

REFERENCES

[1] A. T. Miller and P. K. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.

[2] Y. H. Liu, M. Lam, and D. Ding, "A Complete and Efficient Algorithm for Searching 3-D Form-Closure Grasps in Discrete Domain," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 805–816, 2004.

[3] K. Shimoga, "Robot Grasp Synthesis Algorithms: A Survey," *Int. Journal of Robotic Research*, vol. 15, no. 3, pp. 230–266, 1996.

[4] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," in *IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1824–1829.

[5] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil, "Using Experience for Assessing Grasp Reliability," *Int. Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 671–691, 2004.

[6] C. Borst, M. Fischer, and G. Hirzinger, "Grasp Planning: How to Choose a Suitable Task Wrench Space," in *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 319–325.

[7] G. Biegelbauer and M. Vincze, "Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot's Object Manipulation," *IEEE Int. Conf. on Robotics and Automation*, 2007.

[8] L. Chevalier, F. Jaillet, and A. Baskurt, "Segmentation and Superquadric Modeling of 3D Objects," *Journal of WSCG*, 2003.

[9] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE Int. Conf. on Robotics and Automation*, 2007.

[10] D. Katsoulas, "Reliable Recovery of Piled Box-like Objects via Parabolically Deformable Superquadrics," in *Proceedings of the 9th IEEE Int. Conf. on Computer Vision*, vol. 2, 2003, pp. 931–938.

[11] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander, "Experience based Learning and Control of Robotic Grasping," in *Workshop: Towards Cognitive Humanoid Robots*, 2006.

[12] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, "Early Reactive Grasping with Second Order 3D Feature Relations," in *ICRA Workshop: From Features to Actions*, 2007, pp. 319–325.

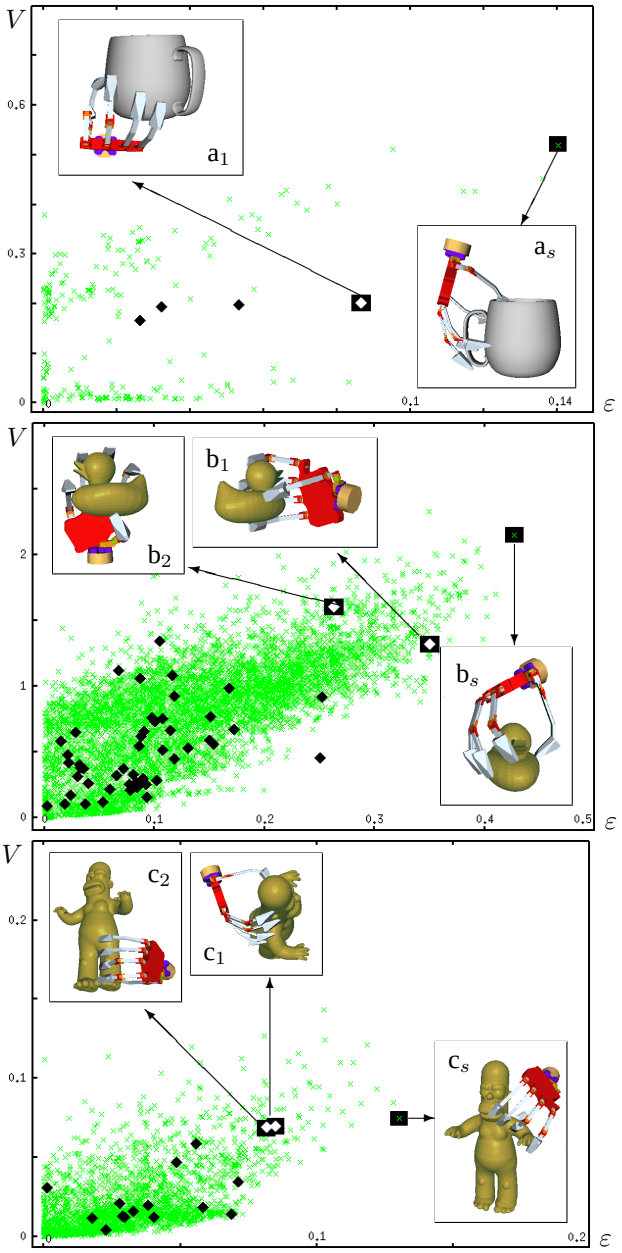


Fig. 6. Grasp quality spaces (ε,V) for models from Fig. 5(a)-(c). Note the depicted samples as a contrast of box grasps and random (spherical) grasps. Legend: × sample of the spherical grasp; ■ best ε-sample of the spherical grasp; ◆ sample of the box grasp; ◆ selected best sample of the box grasp.

[13] H. Zha, T. Hoshide, and T. Hasegawa, "A Recursive Fitting-and-Splitting Algorithm for 3-D Object Modeling Using Superquadrics," in *14th Int. Conf. on Pattern Recognition*, vol. 1, 1998, pp. 658–662.

[14] G. Barequet and S. Har-Peled, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions," *Journal of Algorithms*, vol. 38, pp. 91–109, 2001.

[15] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A Hierarchical Structure for Rapid Interference Detection," *Computer Graphics*, vol. 30, no. Annual Conf. Series, pp. 171–180, 1996.

[16] J.-M. Lien and N. M. Amato, "Approximate Convex Decomposition of Polyhedra," Texas A&M University, Tech. Rep. TR06-002, 2006.

[17] E. L. Damian, "Grasp Planning for Object Manipulation by an Autonomous Robot," Ph.D. dissertation, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 2006.

[18] 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 *Int. Symposium on Robotics (ISR)*, 2006.

Robotics Group
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Technical Report no. 2008 – 1

Tactile Object Exploration using Cursor Navigation Sensors

M. Kjærgaard, D. Kraft, H. Petersen, N. Krüger, A. Bierbaum, T. Asfour and R. Dillmann

January 25, 2008

Title	Tactile Object Exploration using Cursor Navigation Sensors
Author(s)	M. Kjærgaard, D. Kraft, H. Petersen, N. Krüger, A. Bierbaum, T. Asfour and R. Dillmann
Publication History	

Tactile Object Exploration using Cursor Navigation Sensors

M. Kjærgaard, D. Kraft,
H. Petersen and N. Krüger
Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Odense, Denmark
{mortenkj,kraft,hgp,norbert}@mip.sdu.dk

A. Bierbaum, T. Asfour
and R. Dillmann
Institute of Computer Science and Engineering
University of Karlsruhe (TH)
Karlsruhe, Germany
{bierbaum,asfour,dillmann}@ira.uka.de

***Abstract*— In robotic applications tactile sensor systems serve the purpose of localizing a point of contact and measuring contact forces. We have investigated a novel variant of a classic tactile sensor, the Force Sensing Resistor (FSR), which is commonly used in cursor navigation technology. We show the potential of this sensor for active haptic exploration. More specifically, we present experiments and results which demonstrate the extraction of relevant object properties such as local shape, weight and elasticity using this technology.**

An interesting aspect of this sensor is that beside a localization of contact points and measurement of the contact normal force also shear forces can be measured which is relevant for surface normal estimation and weight measurements. Scalable tactile sensor arrays have been developed with this sensor which can be arranged as tiles on a surface, e.g. a manipulator.

I. Introduction

By means of tactile sensing haptic information about an object is acquired during a physical contact between sensor and object. Tactile sensors offer exciting possibilities for use in mechatronic devices and measuring instruments in many areas of science and engineering (see, e.g., [1]).

In this work, we introduce a tactile sensor framework for grasp control and haptic exploration with different robot platforms (e.g., with an industrial robot equipped with a two-jaw gripper and with the humanoid robot platform ARMAR-III [2]) that deploys technology commonly used for cursor navigation on, e.g., laptops. The sensor system is based on available touch sensors involving FSR-technology [3] to acquire the directional contact force vector and the contact location. This type of sensor has originally been developed as cursor navigation input device for hand-held devices. It is therefore low cost and off-the-shelf available. Also, the sensors only need few additional electronic components for embedded integration and there are sufficiently versatile to be applied to manipulators of different geometries.

A comprehensive overview about tactile sensing technology can be found in [1], and more recently in [4] and [5]. It is distinguished between intrinsic sensors, which measure forces internal to the manipulator mechanics, e.g. via load cells at actuation joints, and extrinsic tactile sensors. The latter ones measure forces applied to the manipulator surface and can be found as distributed individual sensors or as dense sensor arrays, which can locate the point of contact on the sensor surface. The contact force itself is derived indirectly by measuring capacity or resistance of the physical sensing element. In Force Sensing Resistors (FSR) [6], a piezoresistive material is used, which varies its electrical resistance in response to an applied mechanical load. Further, there exist also some sensor designs which determine contact force by measuring deformations of the sensor surface with optical sensors [7], [8].

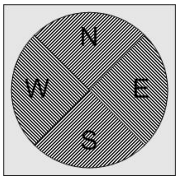
For the purpose of grasp control and shape exploration, measurement of the directional force vector (normal force and shear) and of the contact location is required [9]. The first type of information is usually obtained through load cells in manipulator joints, but it is not possible to also determine the point of contact in multi-contact situations with this type of sensor. For this purpose additional extrinsic tactile sensor arrays are required. New sensor designs for determining both types of information equally are under investigation [10], [11] but have not been shown in an robotic application yet. Further, the latter sensors currently do not provide the dynamic range required in standard robotic grasping or haptic exploration.

We see the technology developed in the context of cursor navigation as an interesting option also for tactile sensing due to its low cost, richness of information (position, normal and shear force) and its modularity. We show the potential of this sensor for active haptic exploration. In particular, we present experiments and results which demonstrate the extraction of relevant object properties such as local shape, weight and elasticity with this technology.

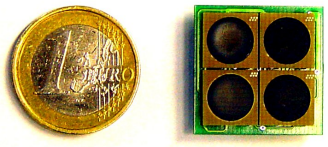
This paper is structured as follows. In the next section the relevant details of the tactile sensor system are described. This includes a description of the sensor characteristics and the proposed calibration method. In section III, we present experiments on the extraction of haptic object properties such as local shape, weight and elasticity. Finally we give a conclusion and an outlook on our future work in section IV.

II. Tactile sensor system

The sensing element of our tactile sensor system is the *MicroNav* cursor navigation sensor from Interlink Electronics [12], which is a four-quadrant FSR sensor. Fig. 1(a) depicts the layout of this sensor element with its four subsensors, labeled N, E, S, W in correspondence to the compass orientations. The sensor element comes in a Surface Mounted



(a) MicroNav sensor layout.



(b) Integrated quadruple sensor array module and 1 Euro coin.

Fig. 1. Tactile sensor.

Device (SMD) package with dimensions $10\times10\times1.4$ mm, the solderable contacts are situated at the bottom side.

The electrical integration is realized with a voltage divider circuit and an Analog-Digital-Converter (ADC) for acquiring the measurement signal as proposed in [6]. Fig. 1(b) shows our realization of a four sensor array module. The sensor module has 16 independent tactile sensing points. By arranging several modules in a dense matrix structure a spatial resolution of 5 mm can be achieved. A microcontroller with integrated ADC, RS232 communication and CAN bus interface is located at the bottom side of the circuit board. With the CAN bus it is possible to interconnect up to 256 individual array boards for realizing a modular tactile sensing system, while the standard RS232 interface is suitable for easily connecting a sensor module to a standard PCs serial interface.

The sensing plane of the *MicroNav* is not supposed to be actuated directly but needs an elastic actuation tip, which both protects the sensor surface and distributes an applied force across the element. For the setup and experiments described we used a rubber actuation tip similar to the reference design [12], see Fig. 2(a).

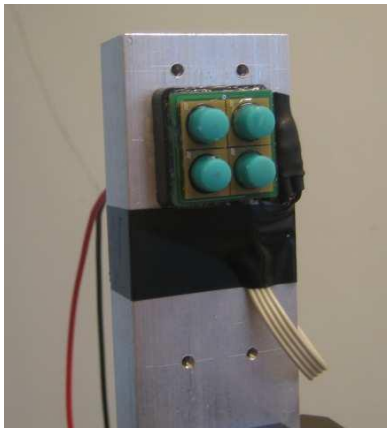
Fig. 2(b) shows an individual *MicroNav* sensor embedded in the silicon rubber actuation tip of a finger in an anthropomorphic hand [13] for the humanoid robot ARMAR-III, a configuration which is still under investigation.

Characteristics of the sensor

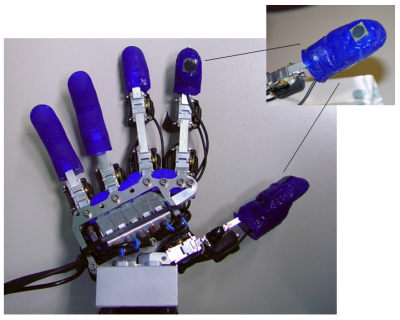
Although FSR sensors are not recommended for precision measurement devices due to their production tolerance ranging from 15% - 25% and their long term settling characteristics, it is possible to calculate a contact force value from the resistance measurement using a calibration procedure.

It should also be noted that FSRs need a minimum force applied for sensing, the so called break force, which limits the measurement range at the lower end. The exact value depends on mechanical characteristics of the sensor and may be adjusted by design of the actuation tip. A typical value for the sensor used is about 0.2 N.

In our calibration setup a single sensor element was mounted to one finger of the parallel gripper of an industrial robot arm. While moving the sensor perpendicular towards the sensitive measurement area of a digital scale, which was used for force measurement here, simultaneous readings of force and sensor output at different pressure levels could be acquired in a measurement sequence.



(a) Sensor array with actuation tips mounted to a gripper.



(b) Single sensor integrated in the finger tip of an anthropomorphic robot hand.

Fig. 2. Different applications of the sensor.

An exemplar measurement of all four subsensors is shown in Fig. 3. It shows that the relationship between force and conductivity of a sensor is not completely linear over the measured range. In the low-force range it is possible to approximate the relationship using a first order function. This will not give the same accuracy as a more complex function but still the result is sufficient for our application of tactile object exploration as we will mainly operate the sensor in this measurement range. It should be noted that since the characteristics of FSR sensors usually differ from part to part, individual calibration is required in general to achieve maximum accuracy.[1]

Because of the intended application measurement values above 4 N will be disregarded in the following. The remaining datapoints were approximated to a straight line using least-squares estimation as illustrated in Fig. 3. It was found sufficient for the application to use a common

[1]Note that the graph representing the W sub-sensor in Fig. 3, is growing clearly faster than the remaining three sub-sensors. Further investigation revealed, that this effect comes from a tangential force component acting upon the sensor tip, which in turn leads to a torque applied to the sensing area. In the future, this problem could be eliminated by reducing height and rotational elasticity of the actuation tip.

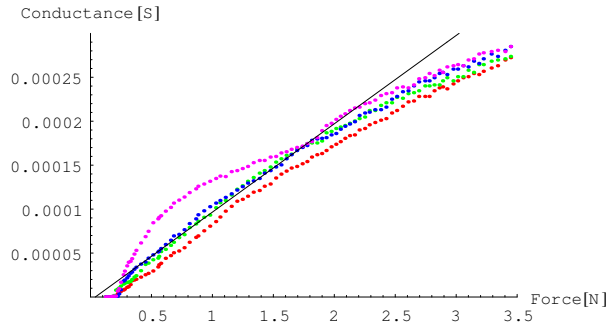


Fig. 3. Measured conductance from all sensors during linearization experiment. Red=N, Blue=S, Green=E, Purple=W

force-conductance relationship for all sensors during our experiments and not to apply individual calibration.

III. Extraction of Haptic Object Properties

For evaluation of the sensor system described above we have performed experiments related to the exploration of various haptic object features. For the experiments one tactile sensor array module was mounted on each finger of the parallel gripper of a *Stäubli Scara* series 6 axis industrial manipulator. For the surface exploration experiments described in Sections III-A and III-B only one finger of the gripper was used, while both fingers were in operation for the experiments in Sections III-C and III-D involving grasping. A dedicated control program on a PC was implemented for each experiment.

The measurement from a sensor element is a four dimensional force vector $\overrightarrow{S}$ consisting of the force measurements from each subsensor

$$\overrightarrow{S} = \begin{pmatrix} n \\ s \\ w \\ e \end{pmatrix} \quad .$$

From this we define a contact force vector

$$\overrightarrow{P} = \begin{pmatrix} n-s \\ w-e \end{pmatrix} \cdot \frac{1}{|\overrightarrow{S}|} \quad .$$

A. Surface normals with single Sensors

The knowledge of the surface normal is an important information in addition to point of contact and force amplitude since it allows for characterizing the shape of objects more precisely. It gives also important information about the stability of a grasp and how to align the grasping device optimally to the object.

In the following experiment we studied the performance of a single *MicroNav* sensor to acquire the orientation of a touched surface, which is directly related to the contact normal force vector.

Instead of describing the contact surface orientation by its normal vector we chose to describe it by two angles, the tilt angle α and the roll angle β, which allows for easier interpretation and qualification of the measurement results.

Fig. 4(b) shows the definition of the tilt angle. A tilt angle of $\alpha = 0$ means the sensor is normal to the surface it is in contact with, and a positive tilt angle means the sensor is tilted towards the N direction. Fig. 4(c) shows the direction of the roll angle. A roll angle of $\beta = 0$ gives a positive tilt in the N direction, $\beta = \frac{\pi}{2}$ gives a tilt in the W direction and so on.

Applying ranges of $\alpha \in \left\{-\frac{\pi}{2}, +\frac{\pi}{2}\right\}$ and $\beta \in \{0, \pi\}$ all possible orientations of a surface relative to the sensor actuation tip can be represented.

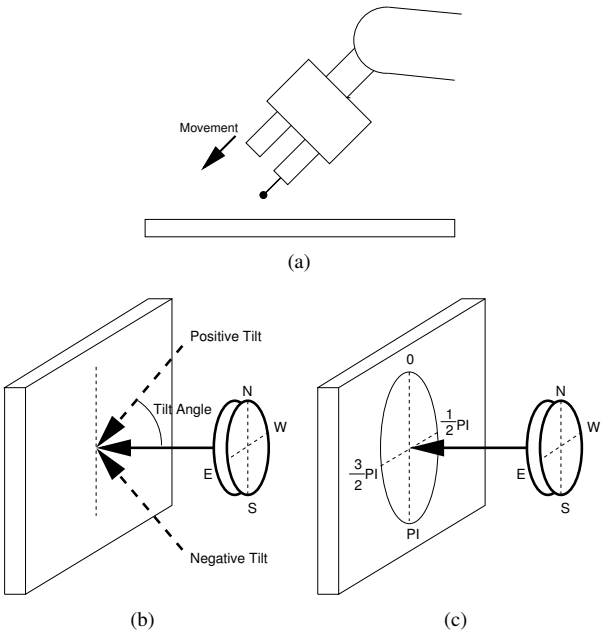


Fig. 4. **(a)** Direction of the robot movement in the experiments. **(b)** Surface orientation defined by tilt angle α. **(c)** Surface orientation defined by roll angle β.

Now all combinations of the following roll and tilt angles were tested by touching a table surface within the workspace of the robot arm:

$$\begin{aligned} \alpha &\in \left\{0, \frac{1}{32}\pi, \frac{2}{32}\pi, \frac{3}{32}\pi, \frac{4}{32}\pi, \frac{5}{32}\pi, \frac{6}{32}\pi\right\} \\ \beta &\in \left\{0, \frac{1}{6}\pi, \frac{2}{6}\pi, \frac{3}{6}\pi\right\} \end{aligned}$$

Every pair of angles was tested six times to collect information about mean value and standard deviation. A graphical representation of the results is shown in Fig. 5, where the $\overrightarrow{P}$-components are drawn versus the applied tilt angle.

The results show that the components of $\overrightarrow{P}$ from the sensor measurement depend on both the roll and the tilt angle. From the measurements the applied tilt angle can be derived up to a value of about $0.4\,rad$ (22°) without becoming ambiguous.

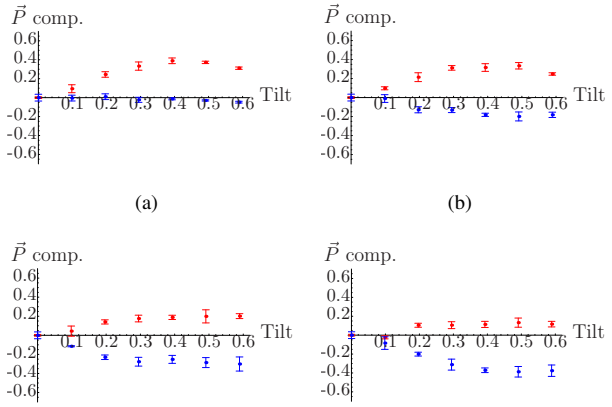


Fig. 5. Results from surface normal experiment using the *MicroNav* sensor. Mean values of $\vec{P}$ shown for the tested tilt angles. The tilt angle is given in radian. Red=North/South axis, Blue=East/West axis. **(a)** For a roll angle $\beta = 0$. **(b)** For a roll angle $\beta = \frac{1}{6}\pi$. **(c)** For a roll angle $\beta = \frac{2}{6}\pi$. **(d)** For a roll angle $\beta = \frac{3}{6}\pi$.

B. Active Surface Exploration

Further, we wanted to investigate the performance of the sensor in shape extraction from haptic data. For this purpose we derived a shape exploration algorithm from the contour follower proposed in [14]. The details of the algorithm are given in the appendix .

For the experiment, a bowl (see Fig. 6) was placed upside down and fixed within the workspace so the robot could press a finger with the sensor array against the surface without moving the bowl. Initially, the controller program needs to be provided with location and orientation of a point on the surface of the bowl as starting point.

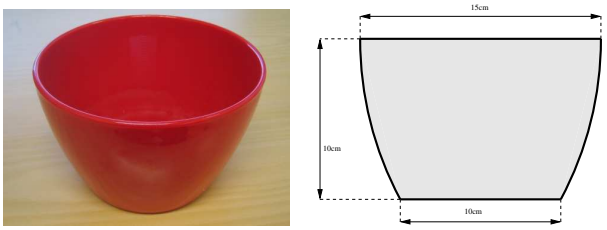


Fig. 6. **a)** The plastic bowl used in the exploration experiment. **b)** The dimensions of the bowl.

During the exploration it was visible that the robot properly aligned the sensor array with the surface. The points found during the exploration movement are illustrated in Fig. 7.

C. Weight

The goal of another experiment was to examine whether the sensor array modules could be used to acquire the weight of an object when grasped by a robot gripper. During the experiment the robot gripper was moved to the specified location of a cup on a table and established a grasp around

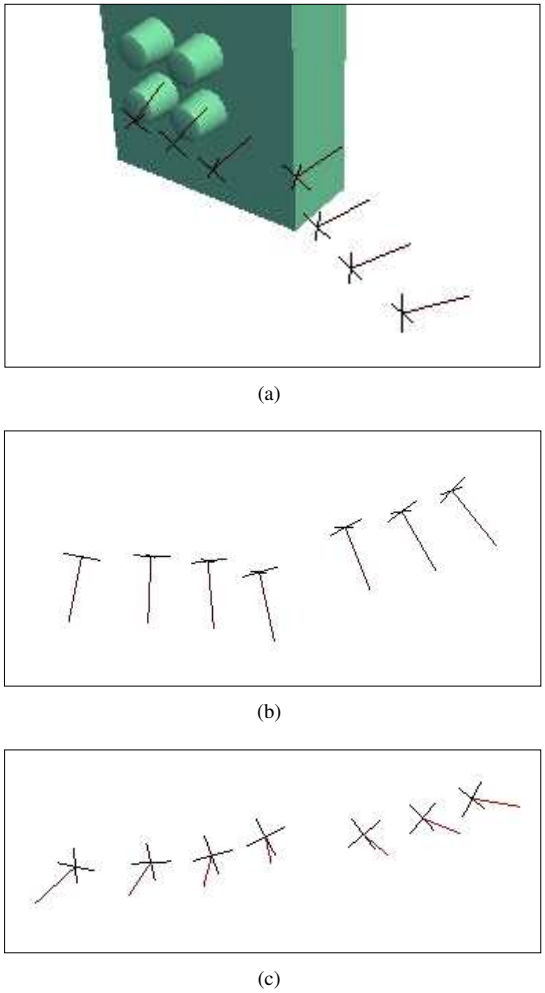


Fig. 7. Surface points found during the exploration of the bowl. The center of the cross marks the position. The red line marks the surface normal. **a)** Robot finger and exploration data as displayed in the control software during exploration. **b)** Surface points and normals seen from top view. **c)** Surface points and normals seen from the front.

it. The exploration procedure realized in the control program was to close the fingers slowly until the sensors could measure a minimum total contact force of 0.5 N from each sensor array, which is enough to provide a stable grasp. From here the object was slowly lifted 1 cm above the surface. The two phases of the experiment are illustrated in Fig. 8(a) and 8(b).

During the lifting phase a torque is applied to the actuation tips by the weight of the object which deforms the elastic material of the tips until an equilibrium is reached when the lifted object has completely left the supporting table.

The sensor values were acquired before and after lifting. The axis through the N- and S-subsensors was aligned to the lifting direction, therefore the difference of the corresponding readings $d = s - n$ was evaluated for examining the influence of the weight on the measurement values. The weight of the cup was increased during several measurements by filling the cup with metal items.

The mean difference over the N-S subsensor pairs of

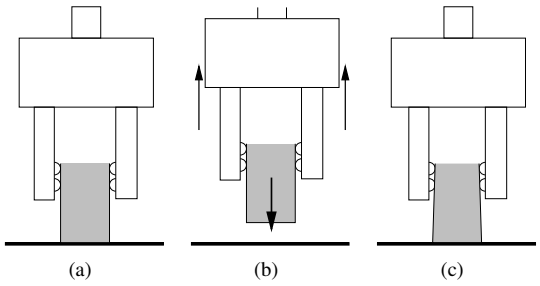


Fig. 8. Feature extraction experiments with the *MicroNav* sensor. **a)** The object is initially placed on the table. **b)** In the weight experiment a grasp is established and the object is lifted. **c)** In the elasticity experiment the object is compressed by the fingers.

all sensor elements for all tested weights is plotted in Fig. 9. The dotted line is a first order approximation to the measurements minimizing the least squares error. The data in this experiment was acquired with a single measurement point for each weight.

For determining the precision we repeated the measurement with a weight of 400 g for 20 times. The standard deviation of this measurement value was found to be 0.22 N.

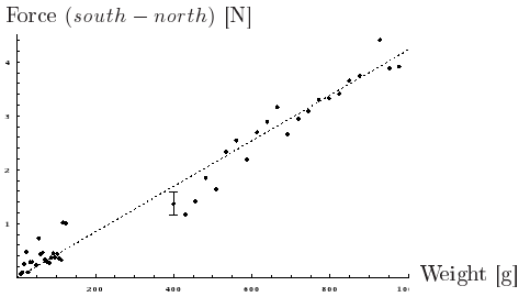


Fig. 9. Results of the *MicroNav* weight experiment

D. Elasticity

In a further experiment we investigated the sensor's ability to discriminate different elasticity values of an object. The same setup as in the preceding experiment was used with a different haptic exploration procedure.

A plastic cup was used as object under investigation, which could be squeezed by the gripper at different heights measured from the cups' bottom, see Fig. 8(c). Naturally, a plastic cup is more rigid close to the cup bottom than to the edge at the top. When pinched at the top, the profile of the plastic cup is deformed from a circular towards an oval shape.

The control program closed the parallel gripper slowly with a constant velocity and stopped when a certain force threshold was exceeded. This experiment was repeated five times at different contact locations along the body of the cup. A plot of the force measurement versus the distance decrement between the gripper fingers is shown in Fig. 10.

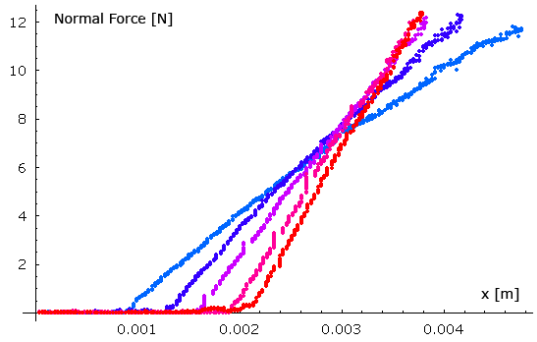


Fig. 10. Results of the elasticity experiment. Light Blue = grasp located at the top, Red = grasp located at the bottom, Other colors = grasp located in between

The plot exhibits a linear relationship which can be interpreted as accordance to Hooke's Law. The spring constant increases linear to larger values for locations closer to the cup bottom, which is reflected in the increasing slope of the plotted lines.[2]

To measure the precision of the elasticity measurements we evaluated the results of multiple measurements (11 times at the top and 16 times at the bottom). The distance traveled by the gripper fingers to reach a threshold force of 5 N is illustrated in Fig. 11. The results for the two measurement points clearly separate. This shows that it is possible to acquire local elasticity of an object using the developed procedure with a parallel gripper.

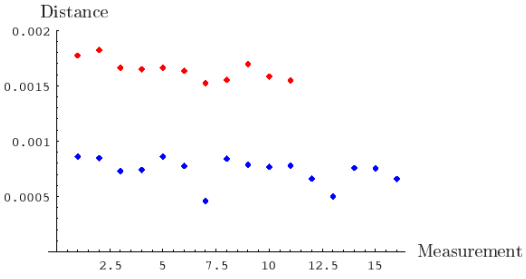


Fig. 11. Results of the *MicroNav* elasticity repeatability experiment. Red dots mark the measurements at the top of the cup. Blue dots mark the measurements at the bottom of the cup.

IV. Conclusions

In this work, we have investigated the potential of a sensor for the purpose of tactile sensing, which has been designed originally in the context of cursor navigation technology. The fact that this sensor is manufactured in mass production makes it cheap (less than 10 Euros per piece) and very robust. In contrast to most existing tactile sensors, it measures not only normal forces but also shear forces which is relevant for a number of applications such as weight measuring, slippage

[2]Note that the lines in Fig. 10 intersect with the x-axis at different coordinates as the diameter at the top edge of the cup is with 59 mm little larger compared to the the bottom with 57.5 mm.

detection, grasp optimization, etc. Also, individual sensors can be mounted in a very modular way to equip rather different grasping devices with tactile sensors. In addition, a high temporal resolution, a decent spatial resolution as well as a wide measurement range are interesting features of this sensor. We have demonstrated the potential of this sensorial framework for three different applications: Surface exploration, weight measurement and elasticity measurement.

As a summary, we believe that the sensors are an interesting alternative to existing tactile sensor systems due to the richness of information they provide, their low price and their modularity.

ACKNOWLEDGEMENT

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

APPENDIX

Our algorithm for shape exploration comprises three phases which are repeated in cyclic sequence:

1) Move robot finger in direction $\hat{n}$, normal to the sensor array, towards the object to be explored and stop when contact is detected, i.e. when the sensor readings exceed a given force threshold.
2) The sensor array must now become aligned with the tangential plane of the surface. During this phase two independent control loops are in operation. For this purpose only the average force readings of each of the four sensor elements are considered. The average force value f_m is calculated as the mean value of all four subsensors for each sensor element respectively. The center point of contact $\overrightarrow{p}$ can be calculated from the geometry of the sensor array and the force readings.
   First, a constant total force f_c, which is measured as the sum of the contact force values, must be maintained in order to keep the applied force of each individual sensor within a specified range. Using a PI velocity controller with coefficients $a_1,\ a_2$ this gives

$$\begin{aligned} e_n &= f_d - f_c \\ v_n &= a_1 e_n + a_2 \int e_n \end{aligned}$$

   with f_d as desired total force and v_n the velocity command in direction normal to the sensor array. This velocity is submitted to the robot arm controller.
   A second controller is required for performing the alignment of the sensor array to the surface normal by rotating around the array center point. The control error $\overrightarrow{e_r}$ is defined as the distance from the contact point location to the center of the sensor array $\overrightarrow{p_c}$. The sensor array is then rotated around the axis perpendicular to the normal vector $\hat{n}$ and $\overrightarrow{e_r}$ with angular velocity $\dot{\theta}$. The corresponding PI controller with coefficients $b_1,\ b_2$ is

$$\begin{aligned} \overrightarrow{e_r} &= \overrightarrow{p} - \overrightarrow{p}_c \\ \dot{\theta} &= b_1 \|e_r\| + b_2 \int \|e_r\| \quad . \end{aligned}$$

   When e_n and $\|e_r\|$ are minimal, the values of p_c and $\hat{n}$ are stored as surface point and normal for this step of the algorithm.
3) The finger is removed from the surface, so that it just releases contact and then moved a short distance in direction tangential to the previously acquired normal vector. From here the algorithm starts again at step 1.

REFERENCES

[1] Mark H. Lee and Howard Nicholls, "Tactile sensing for mechatronics - a state of the art survey," *Mechatronics*, vol. 9, pp. pp.1–31, 1999.
[2] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An Integrated Humanoid Platform for Sensory-Motor Control," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, Dec. 2006, pp. 169–175.
[3] S.I. Yaniger, "Force Sensing Resistors: A Review Of The Technology," in *Electro International, 1991*, April 16-18, 1991, pp. 666–668.
[4] Johan Tegin and Jan Wikander, "Tactile sensing in Intelligent Robotic Manipulation - A Review," *Industrial Robot*, vol. Vol. 32, no. 1, pp. 64–70, No. 1, February 2005, Emerald Group Publishing Limited.
[5] Javad Dargahi and Siamak Najarian, "Advances in tactile sensors design/manufacturing and its impact on robotics applications - a review," *Industrial Robot: An International Journal*, vol. 32, no. 14, pp. 268–281, 2005.
[6] Interlink Electronics, *Force Sensing Resistor Integration Guide*, v1.0 Rev. D, http://www.interlinkelectronics.com.
[7] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 15-19, 2006, pp. 1348–1353.
[8] Jun Ueda, Y. Ishida, M. Kondo, and T. Ogasawara, "Development of the NAIST-Hand with Vision-based Tactile Fingertip Sensor," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 18-22 April 2005, pp. 2332–2337.
[9] Antonio Bicchi, J. Kenneth Salisbury, and David L. Brock, "Contact Sensing from Force Measurements," *The International Journal of Robotics Research*, vol. Vol 12(3), no. 3, pp. 249–262, 1993.
[10] Jong-Ho Kim, Jeong-Il Lee, Hyo-Jik Lee, Yon-Kyu Park, Min-Seok Kim, and Dae-Im Kang, "Design of Flexible Tactile Sensor Based on Three-Component Force and Its Fabrication," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 18-22 April 2005, pp. 2578–2581.
[11] Lucia Beccai, Stefano Roccella, Alberto Arena, Francesco Valvo, Pietro Valdastri, Arianna Menciassi, Maria Chiara Carrozza, and Paolo Dario, "Design and fabrication of a hybrid silicon three-axial force sensor for biomechanical applications," *Sensors and Actuators A: Physical*, vol. 120, no. 2, pp. 370–382, 2005.
[12] Interlink Electronics, *MicroNav Integration Guide*, v3.0, http://www.interlinkelectronics.com.
[13] A. Kargov, T. Asfour, C. Pylatiuk, R. Oberle, H. Klosek, S. Schulz, K. Regenstein, G. Bretthauer, and R. Dillmann, "Development of an anthropomorphic hand for a mobile assistive robot," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, 28 June-1 July 2005, pp. 182–186.
[14] David J. Montana, "The kinematics of contact and grasp," *International Journal of Robotics Research*, vol. 7, no. 3, pp. 17 – 32, June 1988.

Learning Primitive Motions for Robot Poking by Exploration

Damir Omrčen and Aleš Ude

Jožef Stefan Institute
Department of Automatics, Biocybernetics, and Robotics

Pushing, poking, rolling, etc. are examples of nonprehensile manipulation of objects, i.e. object manipulation without a grasp. This kind of object manipulation is used when an object can slip or roll, when an object is too large or too heavy, when an object is out of robot workspace etc. Here we focus on poking as a representative type of nonprehensile manipulation. Poking can be defined as a short term pushing action. Conceptually, our goal is to investigate how to acquire useful action knowledge by observing the results of exploratory actions on objects. For this purpose we study how poking behaviour can be obtained both when the agent randomly pokes the object in different directions and when the agent systematically generates the exploratory pushes to acquire a sufficient amount of examples.

When poking an object, the object motion depends on the object's shape, weight distribution, and on the support friction forces. A lot of work has already been done in the field of mechanics on controllability and planning of poking. Obviously, poking could easily be implemented by assuming a proper representation for the physics of the task, but such an approach relies on a priori knowledge about the action and therefore does not solve the complete learning problem. However, without having physical model of the object and the task, the robot has to experiment with different poking actions on the object. This should enable the robot to acquire knowledge from experimentation and human demonstration in the same way as infants do.

Our work can be divided in two parts. First, the robot needs to learn how an object moves when it is poked from a certain position and from a certain direction. This can be accomplished by experimenting with different poking actions, in which the robot pushes the object several times from different directions and at different locations on the object boundary. During this process the agent builds a knowledge base, which describes the relationship between the point and angle of push on the one side and the actual object movement on the other side. In the second part, the acquired poking knowledge is used to control the object, i. e. to push the object along a prespecified trajectory.

We started by implementing a simulation environment that enables us to study learning and to verify the acquired behaviors. We based our dynamic simulation on ODE (Open Dynamics Engine) library, whose main purpose is to model rigid body dynamics. In our simulation, a pushed object is defined as a planar polygonal object. We have modelled one-point pushing actions by a finger, where a finger is modelled as a cylinder. Figure 1 shows the implemented simulation environment: the polygonal object, the pusher, the direction of the pushing movement (green line) and the resulting object movement (red line).

Simulated learning is realized as follows: the robotic finger performs random (or systematic) poking actions. It performs pokes from different angles and at different

points on the polygon's edge. The system saves points and angles of pokes and the actual directions of object movement. The obtained knowledge has been represented using neural network with two hidden layers. Three different neural networks were used to represent all three directions of movement (two translational and one rotational). Figure 2 shows the response of the object movement in one of the translational directions with respect to the angle and point of contact. The figure left shows the actual movement and the figure right shows the model learned by a neural network.

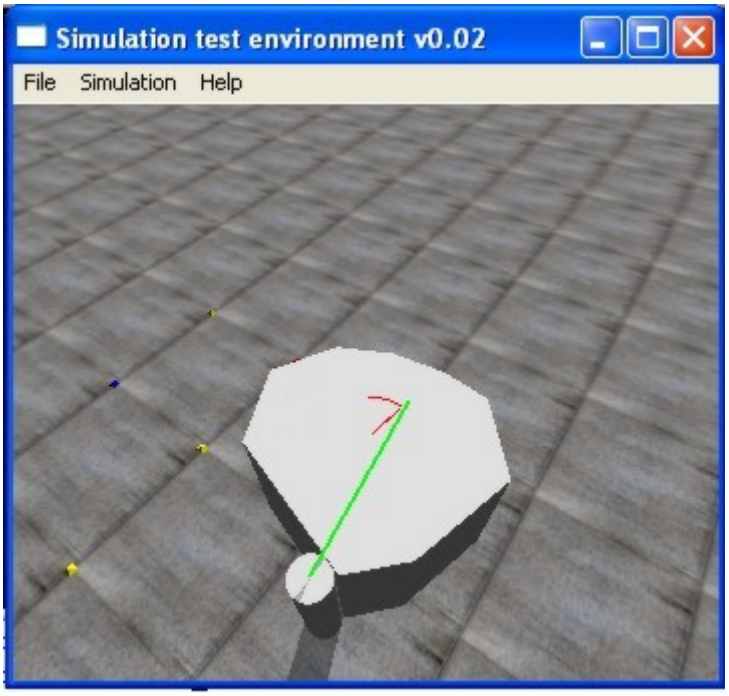


Figure 1: Learning of the pushing behavior by exploration

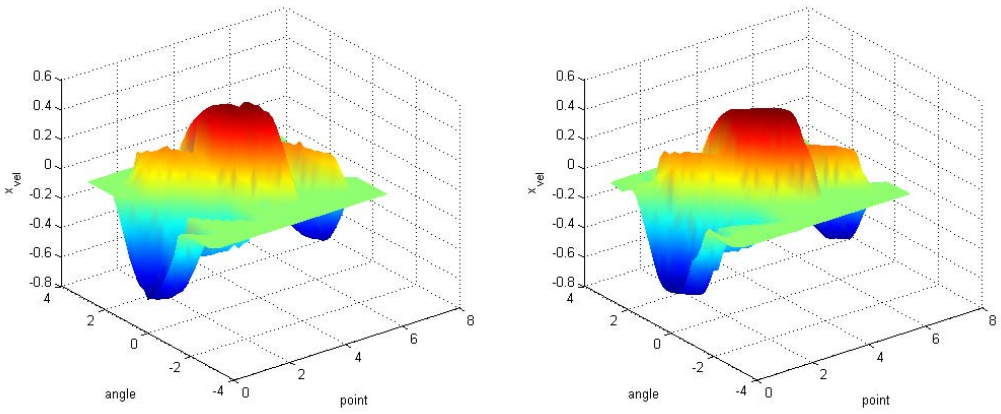


Figure 2: Example of the learned pushing action for *x* direction (left – actual, right – neural network model)

After the learning phase is completed, the agent can use the acquired knowledge to control the object, i.e. to move the object so that it follows a prescribed trajectory. The network has two inputs (point of contact and angle of the pushing action with respect to the normal of the boundary at the point of contact) and three outputs (both translational and the rotational movement). Depending on the desired outputs, we need to find the inputs that produce a suitable movement. Since the inverse system is underdetermined, we have to find the optimal angle and point of push considering other conditions. We have minimized the norm of the error between the predicted and the desired movement. Alternatively, we could ignore the rotational part of motion or we could control the object in such a way to achieve the maximum stability of the pushing action.

After we have defined the desired motion and the system has found the angle and the point of contact, the robot pushes the object. Figure 3 shows the simulation of the pushing behavior. Here, the blue line shows the desired motion and the red line shows the actual motion. Note, that only the direction is important and not the amplitude. We can see that the object motion is close to the desired one.

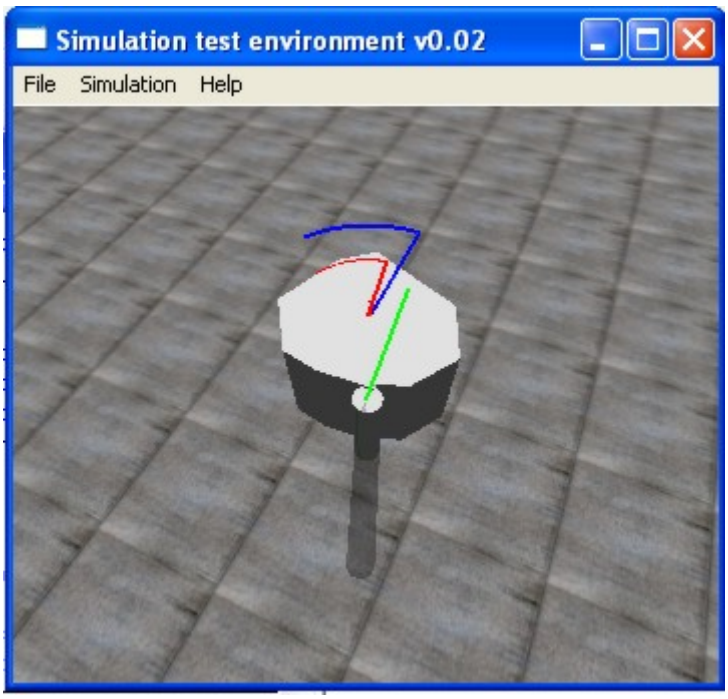


Figure 3: Execution of the pushing action (blue line – desired object motion, red line – actual object motion, green line – pusher movement)

Accurate learning of pushing actions can take a very long time. However, the robot can start using the learned action knowledge even if the learning process has not yet been performed in its entirety. Already after a few explorative poking actions, the agent learns a rough approximation of the relationship between poking and object motion. This initial knowledge can be used for a rather rough control of the object by poking. Next, while controlling the motion the robot can update its knowledge base by observing the actual movement of the object. Thus the relationship between the desired and the actual object motion gradually becomes more precise and the control of the pushing direction gets better.

Learning of poking actions is not accurate enough to be useful without the feedback loop. To push the object along the desired trajectory, it is necessary to modify the point and angle of poke with respect to the actual object motion. We use vision for this purpose. We are currently working on first real robot experiments.

Up to now, we haven't made any generalization of the acquired knowledge. A new object would require us to repeat the learning process. Our future plan is to learn more general models, which will be useful for larger classes of objects. Additionally, to make the learning of poking actions more successful, human instructor can demonstrate the most representative pokes (e.g. perpendicular pokes from a few different sides).

An Early Grasping Reflex in a Cognitive Robot Vision System

Master Theses Report

The Mærsk Mc-Kinney Møller Institute
Faculty of Engineering
University of Southern Denmark

handed in by

Mila Popović

13.12.2007.

Supervisor	:	Asc. Prof. Dr. rer. nat. Norbert Krüger
Co-supervisor	:	Dipl. Inform. Dirk Kraft

Contents

Chapter 1

Introduction

This master theses was completed at the Cognitive Vision Lab, at University of Southern Denmark (SDU). The report presents a novel approach to robotic grasping of unknown objects. It is a guided grasping procedure that is triggered by a stereo visual information. The visual information is processed using the Early Cognitive Vision framework [KLW04] to produce an unique image representation that allows for prediction of objectness. The grasping procedure is a part of a larger cognitive system, where acquiring the physical control over an object is a necessary requirement for learning of an object representation [KBP+].

1.1 Problem statement

In accordance with the idea that embodiment and physical interaction with the environment is a precondition for developing cognition, this work attempts to develop an integrated robot-vision system that performs initial exploration of unknown surroundings.

The theses has following subgoals:

- The realisation of an integrated robot-vision system that implements the early grasping reflex,
- Achieving unsupervised exploration using force torque sensor as an active collision detector,
- The experimental evaluation of the early grasping reflex method in a real environment.

1.2 Outline

The report has the following structure. Chapter 2 gives an overview of the relevant work in the area of grasping unknown objects, followed by an introduction to the approach presented in theses. The hardware and software setup, and the design of the controll application are described in Chapter 3. Chapter 4 presents The Early Cognitive Vision framework and is followed by Chapter 5 that defines the grasping reflex. Active collision detection with force torque sensor is described in Chapter 6. Results of experimental evaluation are presented in Chapter 7. Finally, a conclusion is made in Chapter 8.

Chapter 2

Grasping unknown objects

This Chapter presents a view of the current research in the area of robotic grasping and introduces the grasping approach presented in this work.

2.1 Robotic grasping

Robotic grasping is currently very active research area. Grasping is a key asset for the next generation of service-orientated robots. Another key asset is flexibility - the robots should be able to work in unknown and unstructured environments, be able to grasp and manipulate different objects, deal with uncertainties, work fast, autonomous, safe, reliable, and colaborate and communicate with humans in some intuitive way.

Research in grasping is not new and a significant amount of theoretical knowledge is already available. Analytical approaches [Pet], [BFH04] model interaction between a gripper and an object to investigate properties of the grasp. When contact points between the robot hand and the object are determined and coefficient of friction between the two materials is known, it is possible to calculate a wrench space - 6D space of forces and torques that can be applied by the grasp. A force-closure grasp can resist all object motions provided that the gripper can apply sufficiently large forces.

Grasp planning is a complex problem. Robot hands often have many degrees of freedom and search space off possible grasp configurations is big. Analytical approaches are therefore usually used together with heuristic algorithms. Heuristically-based grasp generators often include some grasp preshape [MKCA03], [Ayd95] types based on human grasping behaviour.

Knowledge based algorithms use the domain specific and a-priori knowledge to reduce the complexity. This knowledge can include workspace constraints, hand geometry, task requirements, perceptual attributes and so on. Other

approaches use learning [PMAJ04], [CPG00], or mimic human pick and place behaviour and learning path [WFG].

2.2 Grasping unknown objects

The research in grasping unknown objects is still at the beginning and varies in respect to complexity. Still most of the projects share this common structure.

- Detecting relevant world features through sensors
- Construction of an approximate object/world model
- Determination and ranking of grasping possibilities
- Execution of the best candidate grasp

The complexity of a system depends on choice of sensors, diversity of admitted objects, the scene configuration, kind of a-priori knowledge allowed and sophistication of the algorithms used. Some projects are focused on working with the limited set of objects or object types. When generic model knowledge is present, the problem is reduced to object recognition of pose estimation.

A number of early projects used visual sensor and a simple gripper with with 2 or 3 fingers [TB94], [BLTK93], [CFMP03]. 2D contour of an object was a relevant feature and grasp planning and quality evaluation was based on approximating the centre of mass of the object with the geometrical centre of the contour. The camera was usually positioned above the scene, pointing vertically down and in some cases several object contours were captured from different angles. Most contemporary vision based approaches assume a simple situation where the scene consists of one object placed against a white background, so that segmentation problem is minimal.

Most newer projects use a range scanning sensors, [TK02], [Ade95], [WJLC05]. It is an apparent choice, since they provide detailed geometrical model of an object. When detailed 3D model is available the grasp planning doesn't differ a lot from the case of grasping known objects.

A research group at Stanford university developed a method for grasping novel objects without the need for building any object model, [SDK+06]. A learning algorithm is used to find the best contact place for grasping an object as a function of an image. The algorithm is trained via supervised learning, using synthetic images as training set. From two or more images that each have their marked "good grasping points" system performs approximate triangulation - to derive 3D position of the grasping point.

2.3 Introduction to grasping reflex

The system presented in this thesis is designed to explore and interact with unknown environment using the minimal amount of a-priori knowledge. The initial information about the world is acquired through stereo vision sensor. The visual information is processed by the Early Cognitive Vision algorithms (Chapter 4) and the acquired world representation triggers a reflex like grasping behaviour of the robot. Early grasping reflex is initial behaviour of a developing cognitive agent.

The premature knowledge of the world is poor and unprecise and cognitive categories of the system are not jet developed. Segmentation of the scene is therefore not possible. Grasping reflex is an autonomous exploration strategy where system produces grasping hypotheses and then tests them by carrying out grasping attempts, in a "trial and error" fashion. The autonomous operation is achieved with help of active collision detection with a force torque sensor mounted the robot wrist in a protected environment. The results of such exploration are later used for constructing the first objects models and represent a basis for development of increasingly complex cognitive categories. The active segmentation is thus achieved through embodied interactions with environment.

The system aims at generating certain percentage of successful grasping hypotheses on arbitrary objects, rather than high quality grasps on a constrained set of objects. In contrast to other approaches that utilise visual sensors, the method described in this report is based on 3D visual representations and grasping is not limited to certain directions. At later stages, that go beyond this report, the system will include tactile sensing and visual attention mechanisms.

Chapter 3

System overview

This Chapter gives a description of the hardware and software elements used in the setup. It also introduces the structure of the grasping procedure and outlines the control application.

3.1 Hardware

The hardware setup consists of a Staubli RX60 six degrees of freedom industrial robot arm, a static Bumblebee2 colour stereo camera, a FTACL 50-80 Schunk Force Torque sensor mounted at the robot's wrist and a PowerCube 2-finger-parallel gripper mounted on the Force Torque sensor, (Figure 3.1). The angular resolution for the six rotational joints of the robot is given in Table 3.1. The floor is covered with flexible foam layer.

The control application is run on a PC machine under Linux operating system. The system uses Modbus interface to communicate to Staubli robot and RS232 serial communication to communicate to the gripper and the force torque sensor. A firewire interface connects the camera to a Windows PC machine that exchange information with the control application through TCP/IP connection.

angular resolution ($^{\circ}10^{-3}$)	0.724	0.724	0.806	1.177	1.953	2.747

Table 3.1: The angular resolution for the six rotational joints of the Staubli RX60 robot. The position repeatability is $\pm$ 0.02 millimetres.

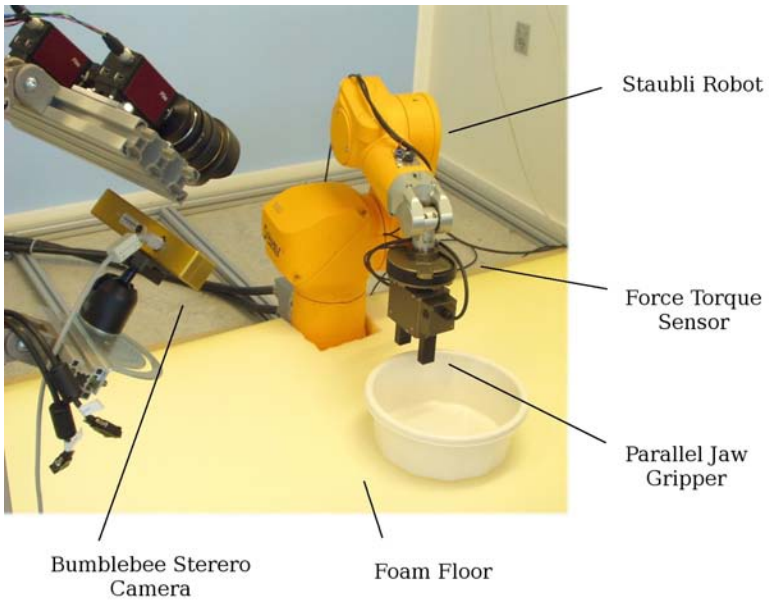


Figure 3.1: Hardware setup elements

3.2 Software

The implementation is based on two distinct software environments CoViS and RobWork. CoViS is a cognitive vision system that is modelling early cognitive functions of biological visual systems, (Chapter 4). It is being developed by the Cognitive Vision Group [CoV] at the University of Southern Denmark. RobWork is a framework for simulation and control of robotics with emphasis on industrial robotics and their applications, [RW]. RobWork environment integrates the Orocos Real-Time Toolkit (RTT), a C++ framework for implementation of (realtime and non-realtime) control systems, [RTT]. Covis and RobWork communicate to each other using a TCP/IP connection.

The image representation reconstructed from CoViS is given in the reference frame of the camera. In order for this data to be used by the robot it has to be transformed into the robot coordinate frame. This is achieved through a robot-camera calibration procedure described in [Kra06], [KW04].

Additionally CoViS provides a visualisation environment WandererX that displays early cognitive vision image representation. A WandererX plugin for displaying grasping hypotheses was developed as a part of this theses.

3.3 Exploration Cycle

This section briefly describes dynamics of one exploration cycle and gives approximate execution times for different steps. In detail description of the state machine will be presented in Chapter 5. One exploration cycle contains following steps:

- The stereo camera captures the images
- Covis processes images and produces a scene representation (Chapter 4)
- The acquired scene representation is used for computing grasping hypotheses (Chapter 5)
- Scene representation and grasping hypotheses (GHs) are loaded into RobWork, where motion planning for grasps is done
- The robot tries to perform certain number of grasping actions
- Actions are accompanied with active collision detection using the force torque sensor (Chapter 6). In case of collision the robot stops, "backs off" to the start position and continues with performing next grasping action.
- When all scheduled actions are performed the system starts from the beginning by taking images of a new scene.

image capturing	$<$ 1 second
image processing	60 seconds
generation of grasping hypotheses	2 seconds
processing of grasping hypotheses	3 seconds
grasping attempt execution	30 seconds

Table 3.2: The approximate run times for different elements that comprise one exploration cycle. These times vary greatly depending on the complexity of the scene.

3.4 Control application

The control application is implemented using the Orocos Real-Time Toolkit (RTT) library and RobWork. It is an extension of the previous application developed for the Cognitive Vision Group by [Kjæ07]. Figure 3.2 outlines the design. The added features are force and torque sensor, active collision detector, and a new state machine that implements grasping reflex exploration cycle and is introduced in Chapter (Chapter 5), (Figure 5.5).

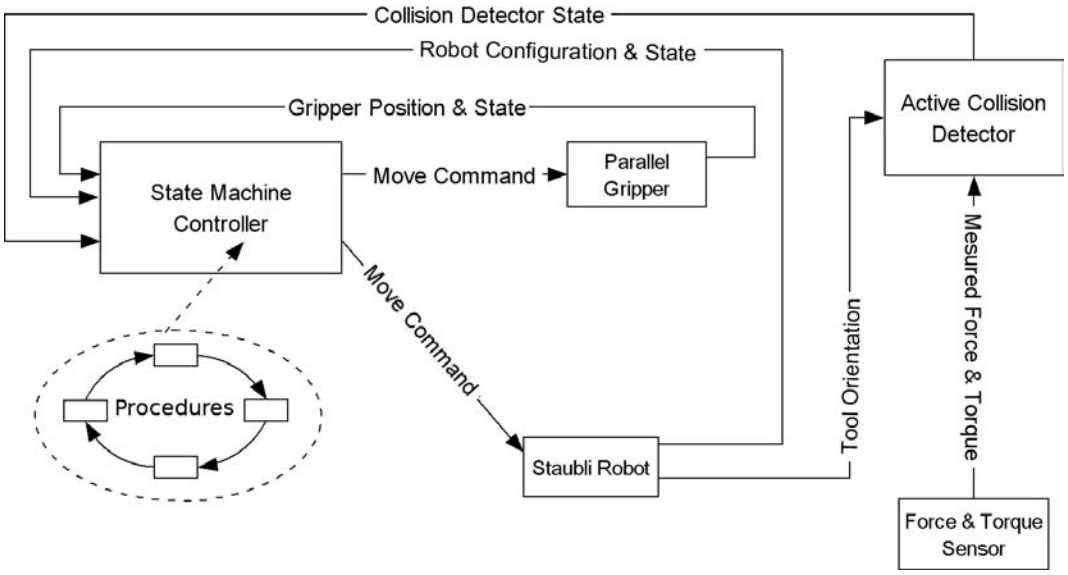


Figure 3.2: Simplified control structure used in application, adapted from [Kjæ07]

Chapter 4

Visual representations

In this Chapter the Early Cognitive Vision framework is introduced. The Chapter starts with describing multi-modal primitives as basic elements of the cognitive vision image representation. The definitions of co-colourity and co-planarity relations between primitives follow. After that the construction of a 3D contour is explained. Last section gives definition of similarity between 3D contours.

4.1 Multi-modal primitives

The early cognitive vision system extracts multi-modal visual features descriptors from stereo images [KLW04]. These visual features are called primitives and give a geometric and appearance based representation of a scene. They are edge descriptors that are extracted sparsely at the points of interest. 2D primitives describe local image patches using different visual modalities such as position of the centre of the patch, orientation of the edge, phase of the signal at this point, colour on both sides of the edge and the local optical flow (Figure 4.1) f. The information contained in 2D primitives is then used for stereo matching (Figure 4.1) g. Resulting 3D primitives can be defined as follows:

$$\Pi = \{\Lambda, \Theta, \Omega, (cl, cm, cr)\}$$

where Λ is the 3D position, Θ is the 3D orientation, Ω is the phase (i.e., contrast transition), and (cl, cm, cr) is the representation of the colour of the spatial primitive, corresponding to the left (cl), the middle (cm) and the right side (cr).

Primitives contain information that allows for the definition of different affinity relations between them such as relations of proximity, collinearity, co-circularity and co-planarity. They can be defined on the 2D and/or the

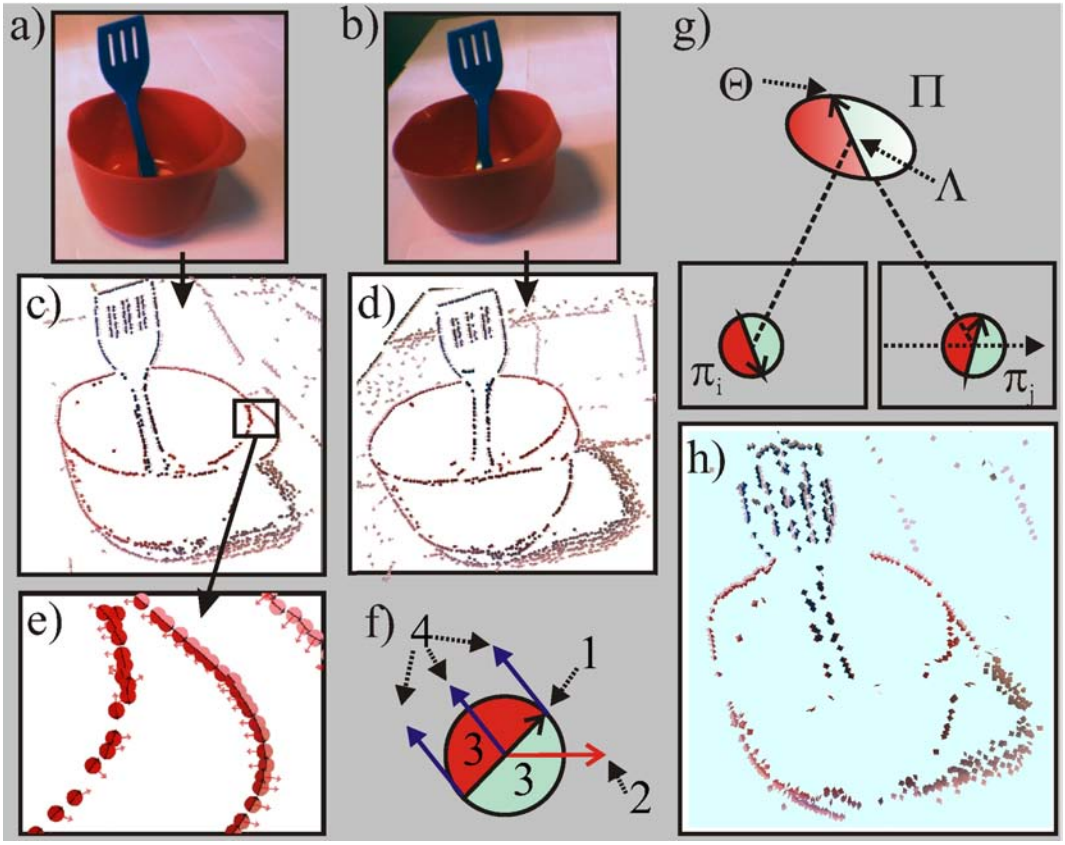


Figure 4.1: Fig. 1. Illustration of 2D and 3D primitives acquired from the vision module. a) and b) show the images captured by the left and right cameras (respectively); c) and d) show the 2D primitives extracted from these two images; in e) a detail of the primitive extraction is shown; f) illustrates the schematic representation of a primitive, where 1. represents the orientation, 2. the phase, 3. the colour and 4. the optical flow. g) from a stereo-pair of primitives (Π_i, Π_j) a 3D primitive Π is reconstructed, with a position in space Λ and an orientation Θ; h) shows the resulting 3D primitives reconstructed for this scenario, From [ASK+07].

3D level and are systematically listed in [KPK07]. Multi-modal affinity relations combine constrains of individual relations to achieve perceptual grouping, [PWK06]. Furthermore, it is possible to establish second order relations between different derived perceptual groups.

Contours represent a simple form of grouping primitives. Relations of co-planarity and co-colourity between contours suggest edges that likely belong to the same object or even the same surface in an unknown scene, and such contours are therefore used for constructing grasping hypotheses. The definitions of co-planarity and co-colourity from [ASK+07] are repeated here for the convenience of the reader, followed by a formal description of 2D and 3D links and 3D contours.

4.2 Co-colourity relation

Figure 4.2 illustrates the co-colourity relation. The co-colourity is computed on 2D level and is true if the sides of the two primitives Π_i and Π_j that are facing each other have the same colour.

$$coc\,(\pi_i, \pi_j) = 1 - \mathbf{d}_c(c_i, c_j) \tag{4.1}$$

where π_i and π_j are the 2D projections of Π_i and Π_j, c_i and c_j are the RGB representation of the colours of the facing parts, and the Euclidian distance between RGB values of the colours c_i and c_j is marked with $\mathbf{d}_c(c_i, c_j)$.

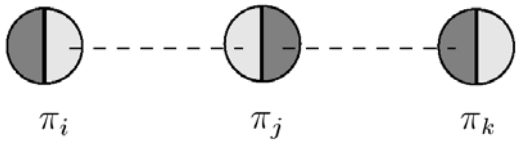


Figure 4.2: Co-colourity of three 2D primitives π_i, π_j and π_k. In this case π_i and π_j are co-colour, so are the π_j and π_k. π_i and π_k are not co-colour, From [ASK+07].

4.3 Co-planarity relation

The co-planarity relation (Figure 4.3) between two spatial primitives Π_i and Π_j is defined as follows. Let $\mathbf{\Lambda}$ be a 3D position of a primitive and $\mathbf{V}_{ij}$ mark a vector connecting the two primitives positions $(\mathbf{\Lambda}_i - \mathbf{\Lambda}_j)$. If $\mathbf{proj_u(a)}$ stands for

$$\mathbf{proj_u}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2} \mathbf{u} \tag{4.2}$$

then the co-planarity relation can be expressed as:

$$cop\,(\Pi_i, \Pi_j) = 1 - |\,\mathbf{proj}_{\Theta_j \times \mathbf{V}_{ij}}(\Theta_i \times \mathbf{V}_{ij})\,| \tag{4.3}$$

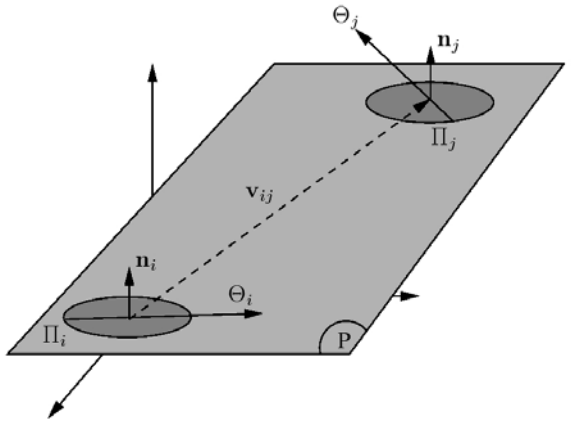


Figure 4.3: Co-planarity of two 3D primitives, [ASK+07].

4.4 2D and 3D links, 3D contours

- A 2D primitive is linked to its neighbour primitive if the "good continuation" constrain is satisfied, i.e. if they are close enough (proximity), collinear or co-circular, and similar in colour, phase and optic flow, [PWK06].
- Two 3D primitives are connected by a 3D link if their corresponding 2D primitives in the left and right image share 2D links.
- 3D contours are formed from 3D links by relation of transitivity, i.e. if a is linked to b, and b to c, then a is linked to c.

Two 3D contours are similar when they are both co-planar and co-colour in the same time. As mentioned above, similar contours are used for constructing grasping hypotheses in an unknown scene. However, extension of co-colourity

and co-planarity relations from 3D primitives to 3D contours is not straightforward. The definition of similarity used in this work is supported by two algorithms. The first one orders primitives inside a contour based on their position (Figure 4.4), and the second is creating associations between primitives of two ordered contours based on their corresponding placement.

Ordering primitives in a contour is accomplished on 2D level in these two steps:

1. The algorithm picks an element e_0 randomly and sorts the other elements based on their distance to e_0, where the distance is Euclidean 2D distance between the centres of the two elements. This step creates only a semi-ordered contour since the beginning or the end of the group is not known. It guarantees to put the most distant element to the end.
2. Repeat step (1) by setting e_0 to the end of the semi-sorted contour.

Two contours C_i and C_j that have N_i and N_j elements are associated in the following way [Kal08]:

1. The two ordered contours are adjusted so that their beginnings, ends and direction of ascend are matching.
2. If N_i equals N_j, the elements are associated in one-to-one manner.
3. If one of the contours, lets say C_i, have smaller number of elements, its elements are associated to approximately C_j/C_i elements of the larger contour C_j, respecting the order.

The similarity of two contours is then defined through similarity of their associated primitives:

If N_s is the number of similar associated primitives, and N_t is the total number of associated primitives then two contours are similar when N_s/N_t is larger than a threshold value.

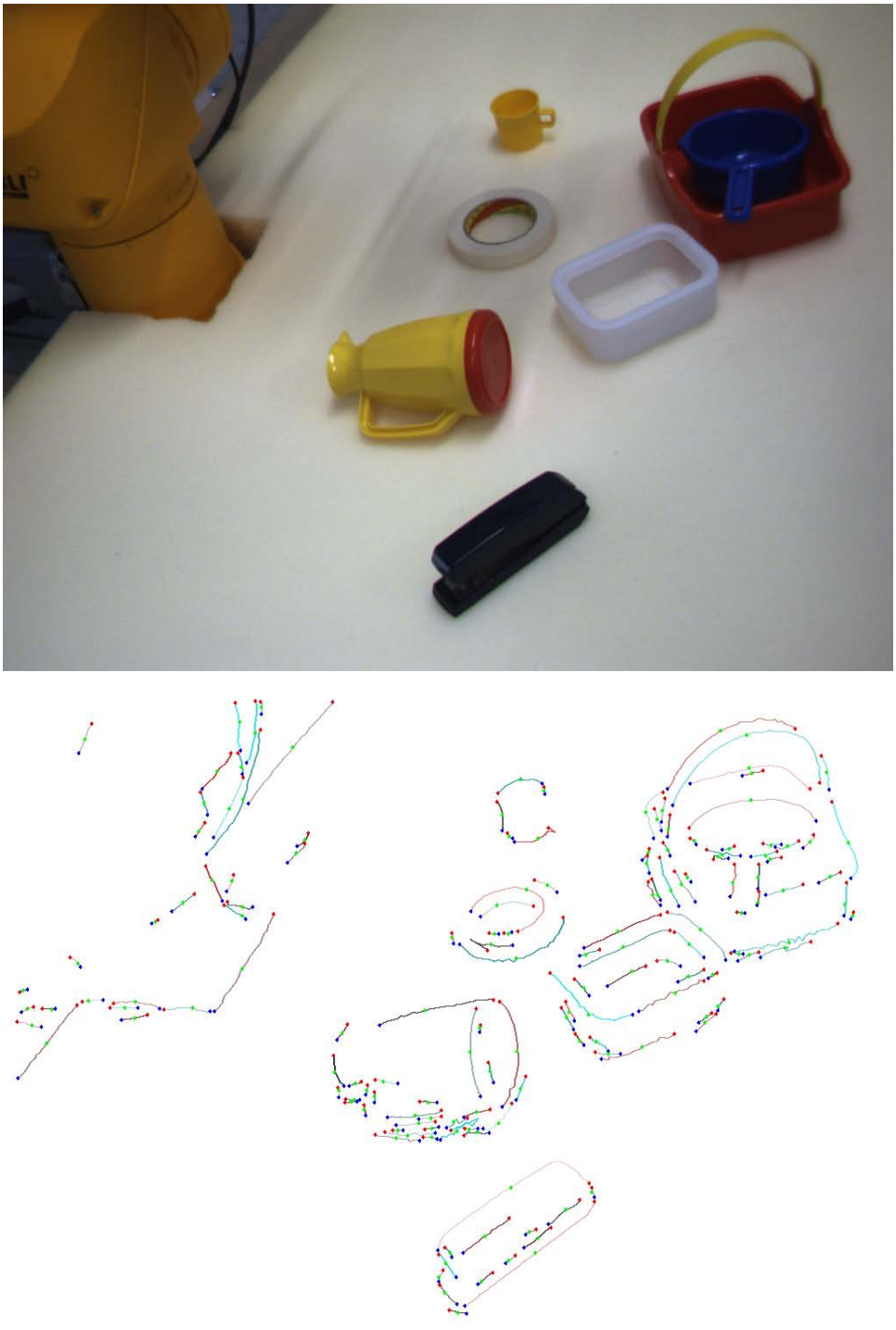


Figure 4.4: Top: A left image from a pair of images captured by the stereo camera. Bottom: Ordered 3D contours extracted from the same scene. Red dots indicate the first primitive in a contour, green the middle, and blue the last primitive in the contour.

Chapter 5

Grasping reflex

Grasping reflex is a low-level procedure that allows for robot manipulator to grasp unknown objects. As explained in Chapter 4, the early cognitive vision system extracts multi-modal visual feature descriptors from stereo images [KLW04]. Multi-modal affinity relations between primitives support perceptual grouping [PWK06]. Second order relations of co-planarity and co-colourity between contours indicate possible co-planar edges originating from the same object, or even the same surface in a scene.

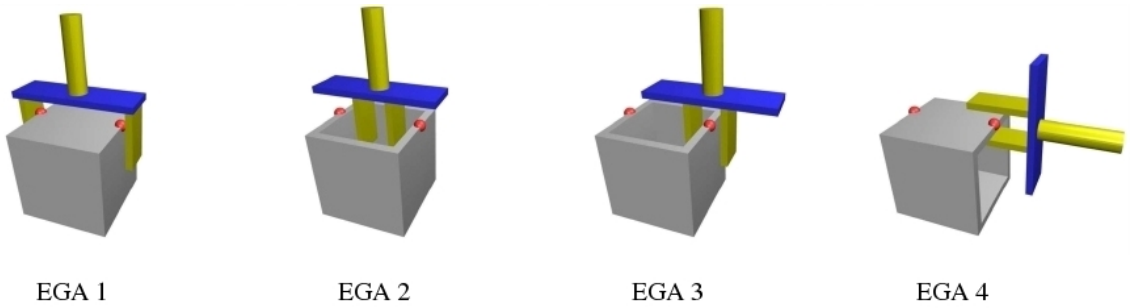


Figure 5.1: Elementary grasping actions (EGAs), adapted From [ASK+07]. The red points indicate 3D primitives that have been reconstructed from stereo image. They appear in pairs, and represent the pair of contours that are connected by relations of co-planarity and co-colourity. In case of EGA 1 and 2 orthogonality to the line connecting the two primitives is required. EGA types 3 and 4 will each generate two actions, one for each parent primitive.

Grasping reflex is based on four basic grasping actions that can be performed on a pair of such edges using a simple parallel gripper. In early cognitive vision edges are represented as 3D contours. As described Chapter 4, 3D contours are sets of the linked 3D primitives. For each of the two similar 3D contours, one representative 3D primitive is chosen. The two primitives are called parent primitives and they carry the information about respective contour's position

and orientation. Figure 5.1 shows the four types of elementary grasping actions (EGAs) defined by the two parent 3D primitives. It is important to notice that in a real scene only some of the four suggested grasps would make sense. For example, if an object on the scene is not a concave object only grasp of type EGA 1 can be successfully performed, provided that other parameters fit. Since the information provided by initial image representation is not sufficient to determine which of the grasping actions are suitable, the system will suggest grasps of all four EGA types. Suggested grasping actions are therefore called grasping hypotheses. The term is also appropriate because grasping actions can fail because of other factors even if the assumed action was reasonable.

The chapter starts by giving mathematical formulation of elementary grasping actions in Section 5.1. After that, Section 5.2 describes in detail algorithm for generating grasping hypotheses for a full scene, the way hypotheses are ranked, processed and performed, and the possible outcomes of a grasping attempts.

5.1 Elementary Grasping Actions (EGAs) definition

The definition of EGAs presented in this work is based on the previous work by [ASK+07] and is slightly modified. Two parent primitives generate up to 6 elementary grasping actions and they belong to one of the four EGA types (Figure 5.1). Their mathematical definition is based on the following set of parameters that are shared for all four grasp types (Figure 5.3):

- Position and orientation of the common plane defined by co-planar parent primitives. It is given by position $\mathbf{P}_p$ and $\mathbf{n}_p$ orientation of the plane normal
- Direction connecting the parent primitives - $\mathbf{D}$
- distance between parent primitives - d_p
- individual primitives orientations Θ_i and Θ_j

The grasp itself is defined with position and orientation of the the tool expressed in the Robot's Base reference frame, (Figure 5.2) and with initial finger distance d. The position is defined as a 3D position of the Tool Centre Point (TCP) reference frame. TCP is positioned between gripper fingers, the z distance of the TCP from the gripper finger's ends gives how "deep" the grasp is. Orientation is given by two TCP main axis directions, which is enough to define the desired rotation of the TCP frame expressed in the Robot's Base frame, the third direction is then calculated as the cross product of the other two. Z axis of TCP frame $\mathbf{Z}_{TCP}$ is parallel to the gripper's fingers, $\mathbf{X}_{TCP}$ axis connects the fingers, and $\mathbf{Y}_{TCP} = \mathbf{Z}_{TCP} \times \mathbf{X}_{TCP}$.

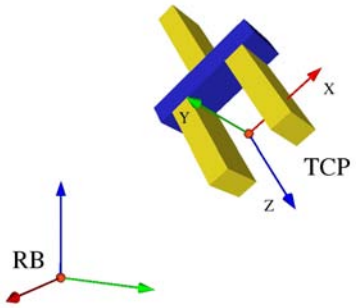


Figure 5.2: The Figure shows the Tool Centre Point (TCP) reference frame, it is given in respect to the robot's base (RB) frame. The position and orientation of the TCP reference frame is used when defining elementary grasping actions.

common plane definition

If a pair of primitives is represented with $\{\Pi_i, \Pi_j\}$, and position and normalised orientation of a primitive Π are respectively represented with $\Lambda(\Pi)$ and $\Theta(\Pi)$ then it can be calculated:

$$\mathbf{D} = \frac{\Lambda(\Pi_j) - \Lambda(\Pi_i)}{\| \Lambda(\Pi_j) - \Lambda(\Pi_i) \|}$$

$$\begin{aligned} \mathbf{n}_i &= \Theta(\Pi_i) \times \mathbf{D} \\ \mathbf{n}_j &= \Theta(\Pi_j) \times \mathbf{D} \end{aligned} \tag{5.1}$$

where $\mathbf{D}$ is the direction of the vector connecting the two primitives, $\mathbf{n}_i$ and $\mathbf{n}_j$ are the normals to the planes that each of the primitives defines in relation to the vector connecting them (Figure 5.3a). The two planes are combined to form one common plane p defined by the position $\mathbf{P}_p$ and the direction $\mathbf{n}_p$ of its surface normal.

$$\mathbf{P}_p = \Lambda(\Pi_i) + \frac{\Lambda(\Pi_j) - \Lambda(\Pi_i)}{2}$$

$$\mathbf{n}_{p\,1/2} = \pm \frac{\mathbf{n}_i + sw \cdot \mathbf{n}_j}{\| \mathbf{n}_i + sw \cdot \mathbf{n}_j \|} \tag{5.2}$$

$$sw = \begin{cases} -1 & \text{if } \mathbf{n}_i \cdot \mathbf{n}_j < 0 \\ 1 & \text{otherwise} \end{cases}$$

Switch factor ("sw")

The two line segments generating a grasp are generally not perfectly co-planar (Equation 4.3, Figure 5.3(a)). It follows that $\mathbf{n}_i$ and $\mathbf{n}_j$ define two different planes (Equations 5.1). For generating elementary grasping actions, one plane is used. Equation 5.2 acts as an averaging operator for the two planes. The switch factor assures that the averaging is optimal. Since both $\mathbf{n}_i$ and $\mathbf{n}_j$ are orthogonal to $\mathbf{D}$, the role of the switch factor can be illustrated in 2D. It is important to mark that the directions of the surface normals are arbitrary, as line segments orientation vectors have arbitrary directions and because they are depending on the relative orientation to the connecting direction $\mathbf{D}$. Figure 5.3(b) shows the case where normals point in similar directions, and the addition of the two normals will correctly determine a plane in between the two planes. In the case where derived normals point in the opposite directions and an addition is performed, the resulting plane will be orthogonal to the optimal solution, 5.3(c). The two normal vectors have to be subtracted instead.

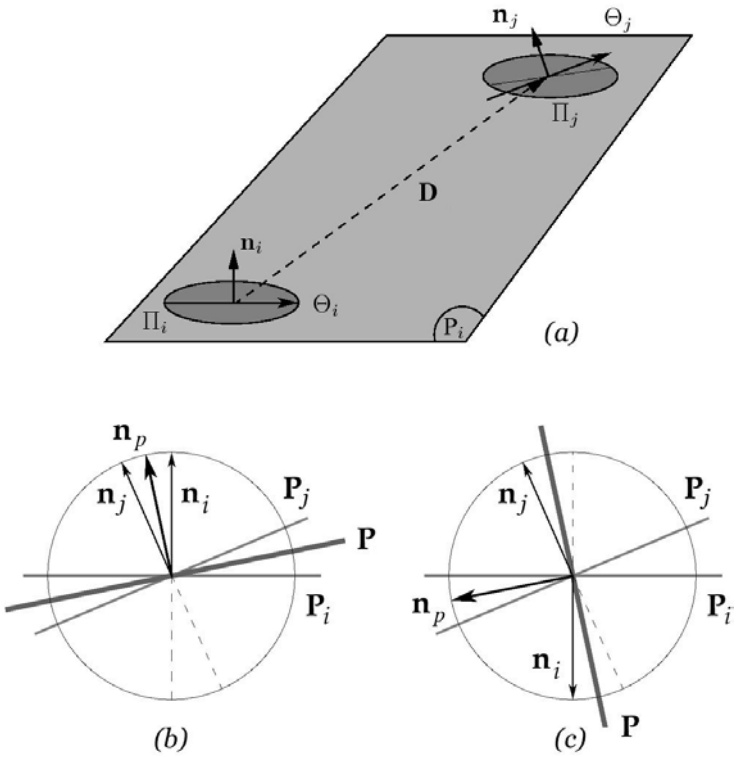


Figure 5.3: Calculation of the common plane between two co-planar 3D primitives (Equation 5.2). Figures (b) and (c) illustrate the use of the switch factor.

Choosing normal direction

The plus-minus sign on the righthand side of the Equation 5.2 indicates that the direction of the normal of the averaged plane is arbitrary. The early grasping reflex is a guess on how to grasp the two co-planar edges (Figure 5.1). Each of the two corresponding 3D contours give one representative 3D primitive, and they are used for calculation of the common plane and the other grasp parameters. It is important to know which direction of the plane normal to use in order to predict meaningful grasps. The initial scene representation doesn't provide this information. Nevertheless, it is intuitively clear to the human viewer why the top side of the box on Figure 5.1 (EGA1) should be grasped from above. This observation can be expressed mathematically. The normal of the visible side of a surface always forms an obtuse angle to the vector originating from the point of view and pointing to the surface normal.

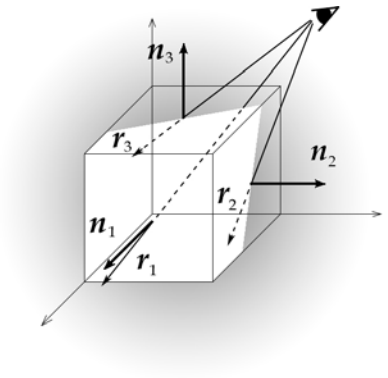


Figure 5.4: Choosing the correct surface normal. $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ are outward surface normals marking the sides of the cube visible on the illustration. $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are camera rays, vectors originating from the marked point of view and pointing to the surface normals.

$\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ are normals of three cube surfaces (Figure 5.4). Directions chosen are ones that would create correct grasping hypotheses. However, only the surfaces 2 and 3 are visible from the marked point of view. $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are camera rays. If a surface normal and a ray pointing from the camera to that plane form an obtuse angle ($\mathbf{n} \cdot \mathbf{r} < 0$), the surface is visible from the camera. If the angle is acute ($\mathbf{n} \cdot \mathbf{r} > 0$) the surface is not visible from the camera. When this observation is turned around, it follows that visible surfaces should adopt the direction of the normal that forms an obtuse angle to the camera ray in order to give expectable grasps. This is not an optimal solution because surfaces that are not visible from the camera's point of view can still display some edges. In the given example (Figure 5.4) the cube's invisible surfaces can display up to two edges. Those edges also suggest grasping hypotheses. Still, in the majority of cases determining a normal direction in this way is better

then having two possible normal orientations as this would produce two sets of grasping hypotheses, and many more wrong hypotheses altogether. The same reasoning is applicable in EGA2 and EGA3 (Figure 5.1) cases, when grasping co-planar edges of a concave object. EGA3 type of grasp doesn't depend on plane normal direction. Another aspect of this observation concerns camera placement. Visible features of objects should be the ones reachable by the manipulator.

EGAs mathematical formulation

EGA 1 is a grasp where the gripper aims at holding a whole object, the fingers of the gripper are initially wide open and the object is grasped at its full width by closing fingers, (Figure 5.1 EGA 1). The position of the origin of TCP - $\mathbf{P}_{TCP}$ equals the position of the common plane (defined as a middle point between the parent primitives). Gripper's $\mathbf{Z}_{TCP}$ direction is aligned with common plane direction, but has the opposite sign. The $\mathbf{X}_{TCP}$ direction (direction connecting fingers) is identical to the direction of the line connecting parent primitives. Initial finger distance d should be bigger then distance between parent primitives d_p, so that grasping position can be approached without colliding with the object. It is limited by the maximum fingers opening distance d_{max}.

$$\begin{aligned} \mathbf{P}_{TCP} &= \mathbf{P}_p \\ \mathbf{Z}_{TCP} &= -\mathbf{n}_p \\ \mathbf{X}_{TCP} &= \mathbf{D} \\ d_p < d &\leq d_{max} \end{aligned} \tag{5.3}$$

where $\mathbf{P}_p$ is position of the common plane normal, $\mathbf{n}_p$ is the normal and $\mathbf{D}$ is the direction of the connection line between primitives. The $\mathbf{X}_{TCP}$ could have opposite direction as well ($-\mathbf{D}$), as the gripper has 180° symmetry around z axis in respect to grasping. This fact will be taken into account later when deciding optimal grasp robot configurations, (Section 5.2).

EGA 2 is a grasp that is designed for concave objects, it has same position and orientation as EGA 1 but the initial finger distance is zero and finger are opened in order to grasp an object, (Figure 5.1 EGA 2).

$$\begin{aligned} \mathbf{P}_{TCP} &= \mathbf{P}_p \\ \mathbf{Z}_{TCP} &= -\mathbf{n}_p \\ \mathbf{X}_{TCP} &= \mathbf{D} \\ d &= d_{min} \end{aligned} \tag{5.4}$$

Since the grasping tool is a simple parallel gripper EGA 1 and 2 grasps will be successful only when parent primitives individual orientations are co-planar and orthogonal to the line connecting them, meaning that the two parent co-planar contours should be parallel and the two representative primitives should be opposite to each other. If this is not the case the grasp is unstable or not possible. The parent primitives are co-planar and the orthogonality to the connecting line is defined as in Equation 5.5, where C is a positive real number smaller then one.

$$|\Theta(\Pi_i) \cdot \mathbf{D}| < C \quad \wedge \quad |\Theta(\Pi_j) \cdot \mathbf{D}| < C \tag{5.5}$$

EGA 3 type of grasp is also designed for a concave object, where concavity is placed between two defining edges. The gripper tries to grasp an object by holding one if its sides, (Figure 5.1 EGA 3). As there are two parent primitives (two edges), two grasping actions will be generated for the same parent primitives pair. $\mathbf{P}_{TCP}$ is coinciding with position of the parent primitive. $\mathbf{Z}_{TCP}$ direction is the same as in EGA 1 and 2 cases, but $\mathbf{Y}_{TCP}$ is used as other defining direction and is based on the individual orientation of the parent primitive. This is why for EGA 3 grasps the orthogonality to the connecting line is not a requirement. $\mathbf{Y}_{TCP}$ direction is calculated as normalised projection of the primitive direction to the common plane.

The projection $\mathbf{a}_s$ of a direction vector $\mathbf{a}$ to a plane s is calculated by projecting the vector to the plane normal $\mathbf{n}_s$ and then subtracting that projection $\mathbf{a}_n$ from the original direction vector. Normalised projection is labelled with $\hat{\mathbf{a}}_s$.

$$\begin{aligned} \mathbf{a}_n &= (\mathbf{a} \cdot \mathbf{n}_s) \cdot \mathbf{n}_s \\ \mathbf{a}_s &= \mathbf{a} - \mathbf{a}_n \\ \hat{\mathbf{a}}_s &= \frac{\mathbf{a}_s}{\| \mathbf{a}_s \|} \end{aligned} \tag{5.6}$$

Normalised projection of the primitive's orientation to the common plane p is marked with $\hat{\Theta}(\Pi_i)_p$. Initial finger distance is decided so that the gripper finger that is placed on the inner side of the edge has equal distance to both parent edges, it is a function of partent primitives distance d_p and finger thickness ft. If $d_p - ft > d_{max}$ then d is reduced to d_{max}.

$$\begin{aligned} \mathbf{P}_{TCPi} &= \Lambda(\Pi_i)\,,\ i = (1,2) \\ \mathbf{Z}_{TCP} &= -\mathbf{n}_p \\ \mathbf{Y}_{TCP} &= \hat{\Theta}(\Pi_i)_p \\ d &= min(\, d_p - ft,\ d_{max}\,) \end{aligned} \tag{5.7}$$

where Π_i is one of the two primitives, $\Lambda(\Pi_i)$ is the position of the primitive and $\Theta(\Pi_i)$ is the orientation of the primitive.

In EGA 4, the gripper is trying to grasp a surface defined by two parent primitives, (Figure 5.1 EGA 4). Two grasps are possible, one from the each parent. The distance between parent primitives can not be used for determining initial finger distance, and the value is assigned as constant. For most objects, approaching the grasp is easiest when gripper has maximum opening initially, but for smaller objects this might lead to collision of the gripper finger with the other edge. As for EGA 3 grasps $\mathbf{P}_{TCP}$ matches the position of the parent primitive. The computation of the orientation includes all three directions $\mathbf{X}_{TCP}$, $\mathbf{Y}_{TCP}$, and $\mathbf{Z}_{TCP}$. $\mathbf{X}_{TCP}$ equals to $\hat{\Theta}(\Pi_i)_p$, $\mathbf{Y}_{TCP}$ is $\pm\mathbf{n}_p$ and $\mathbf{Z}_{TCP} = \mathbf{X}_{TCP} \times \mathbf{Y}_{TCP}$. Compact definition of EGA 4 is

$$\begin{aligned} \mathbf{P}_{TCPi} &= \Lambda(\Pi_i)\,,\ i = (1,2) \\ \mathbf{X}_{TCP} &= \pm\mathbf{n}_p \\ \mathbf{Y}_{TCP} &= \hat{\Theta}(\Pi_i)_p \\ d &= constant \end{aligned} \tag{5.8}$$

The correct sign in front of the normal $\mathbf{n}_p$ insures that $\mathbf{Z}_{TCP}$ has such orientation that would make the gripper fingers point to the surface between two parent primitives. It is chosen in the following way. Let us assume the solution:

$$\begin{aligned} \mathbf{X}_{TCP} &= \mathbf{n}_p \\ \mathbf{Y}_{TCP} &= \hat{\Theta}(\Pi_i)_p \\ \mathbf{Z}_{TCP} &= \mathbf{X}_{TCP} \cdot \mathbf{Y}_{TCP} \end{aligned}$$

If $\mathbf{Z}_{TCP} \cdot \mathbf{D} > 0$ (in case of primitive i), or $\mathbf{Z}_{TCP} \cdot \mathbf{D} < 0$ (for primitive j), where $\mathbf{D}$ is the direction of the vector connecting two parent primitives, defined as in Equation 5.1, Figure 5.3a, then solution above is accepted. If $\mathbf{Z}_{TCP} \cdot \mathbf{D} < 0$ (primitive i), or $\mathbf{Z}_{TCP} \cdot \mathbf{D} > 0$ (primitive j), the opposite sign of the normal is chosen and the $\mathbf{Z}_{TCP}$ is recalculated:

$$\begin{aligned} \mathbf{X}_{TCP} &= -\mathbf{n}_p \\ \mathbf{Z}_{TCP} &= \mathbf{X}_{TCP} \cdot \mathbf{Y}_{TCP} \end{aligned}$$

5.2 Grasps generation, processing and testing

Figure 5.5 shows the state diagram for the grasping reflex procedure. Grasping hypotheses generation starts with acquiring stereo images of a scene and

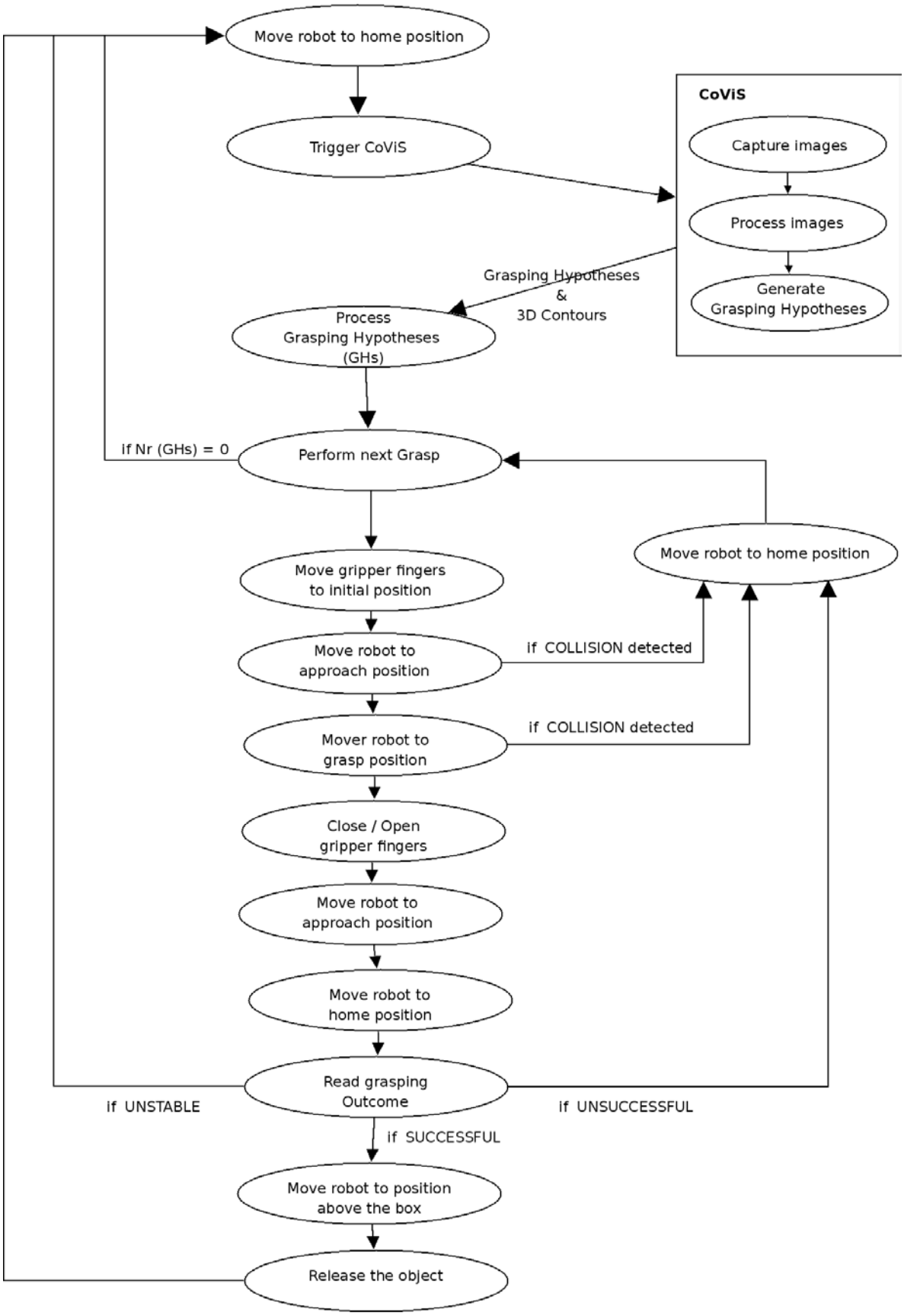


Figure 5.5: State diagram for grasping reflex exploration.

producing a cognitive vision image representation. Position and orientation of the 3D primitives is smoothed [PWK06]. From image representation number of 3D contours are extracted and ordered, where a 3D contour contains minimum three 3D primitives. All 3D contours are then compared with each other for similarity and a list of pairs of similar contours is made. For each similar contour a representative 3D primitive is chosen as a corresponding middle primitive.

3D positions of both representative primitives in a similar contour pair have to be inside a certain region of interest before the pair is excepted as a parent primitives pair for generating grasping actions. This region of interest is defined as a bounding box in front of the robot and is used to exclude contours originating from robot's or background features.

One pair of parent primitives can produce at most six grasping hypotheses, one for each EGA1 and EGA 2 and two for each EGA 3 and EGA 4 type of grasp. The number of produced hypotheses depends on different conditions. EGA 1 grasp will be produced if parent primitives are orthogonal to the line connecting them (Equation 5.5), and if distance between parent primitives d_p is inside limits:

$$5mm \leq d_p \leq 58mm$$

Parent primitives with distance smaller than the lower limit are most likely belonging to the same edge. The upper limit equals to the maximum finger distance (68mm) minus 10 mm. This means that the gripper fingers have minimum 5 mm distance (on each side) to the object during approaching.

EGA 2 grasp is produced if parent primitives are orthogonal to the connecting line and if parent primitives distance satisfies following conditions:

$$50mm \leq d_p \leq 108mm$$

Lower limit means that the opening between two parent primitives has to be at least the width of two fingers (2 x 20mm) plus 10 mm so that the gripper can approach initial position. If the distance is bigger then two fingers width plus the maximal distance between fingers, the outer sides of fingers can not reach the edges of the object and can therefore not apply any force.

In case of EGA 3 there is only one lower limit that assures that one finger can fit the opening of the concave object. There is no upper limit to parent primitives distance. The lower limit equals to one finger width plus 10 mm.

$$30mm \leq d_p$$

EGA 4 will not make sense unless the parent primitives belong to two distinct co-planar parent contours, that define and bound a surface:

$$5mm \leq d_p$$

Grasp reachability and collision checking

Not all of the grasping hypotheses produced by the image representations are reachable by the robot. Staubli RX60 robot has six degree of freedom, which means that six independent joint angles have to be specified in order to fully describe robot's configuration [Stä05]. Because of many degrees of freedom, a specific position and orientation of the gripper can be achieved with more than one robot configuration. A problem of finding all possible sets of joint angles that lead to specific tool position and orientation is called inverse kinematics. A closed form Pieper's [Cra89] solution is available for Staubli RX60 robot and is used in this work.

Even when a grasp is reachable by the robot, it might be inaccessible because some part of the robot's body, (including the force torque sensor and the gripper mounted on the robot's wrist), might collide with environment or with itself. Initial knowledge of the environment is limited to knowledge of robot's own geometry and the position of the ground plane. Additional information becomes available when vision system produces image representation from pair of stereo images. 3D contours from image representation corespond to edge features in the scene. RobWork simulation environment uses geometric and kinematic model of the robot, and geometric models of the floor and 3D contours to check if certain robot configurations are collision free, using PQP (Proximity Query Package) [LGLM99] collision detection strategy. 3D contours are sets of 3D primitives. Each 3D primitive is modeled as a small cube (Figure 5.7).

Choosing grasping hypotheses

All grasping hypotheses from all similar contour pairs in a certain region of interest compose a full set of grasping hypotheses produced from the image representation. The size of the set usually varies from several to several thousand grasping hypotheses, depending on the scene complexity and the quality of the reconstruction. When a grasping attempt is performed, a robot is affecting the environment and it is possible that the scene will change, and thus some of the grasping hypotheses will no longer be valid. The system will therefore, after a few grasping attempts, start a new cycle of capturing images, extracting image representation and generating grasping hypotheses is started. In order to increase effectiveness of exploration it is necessary to carefully chose

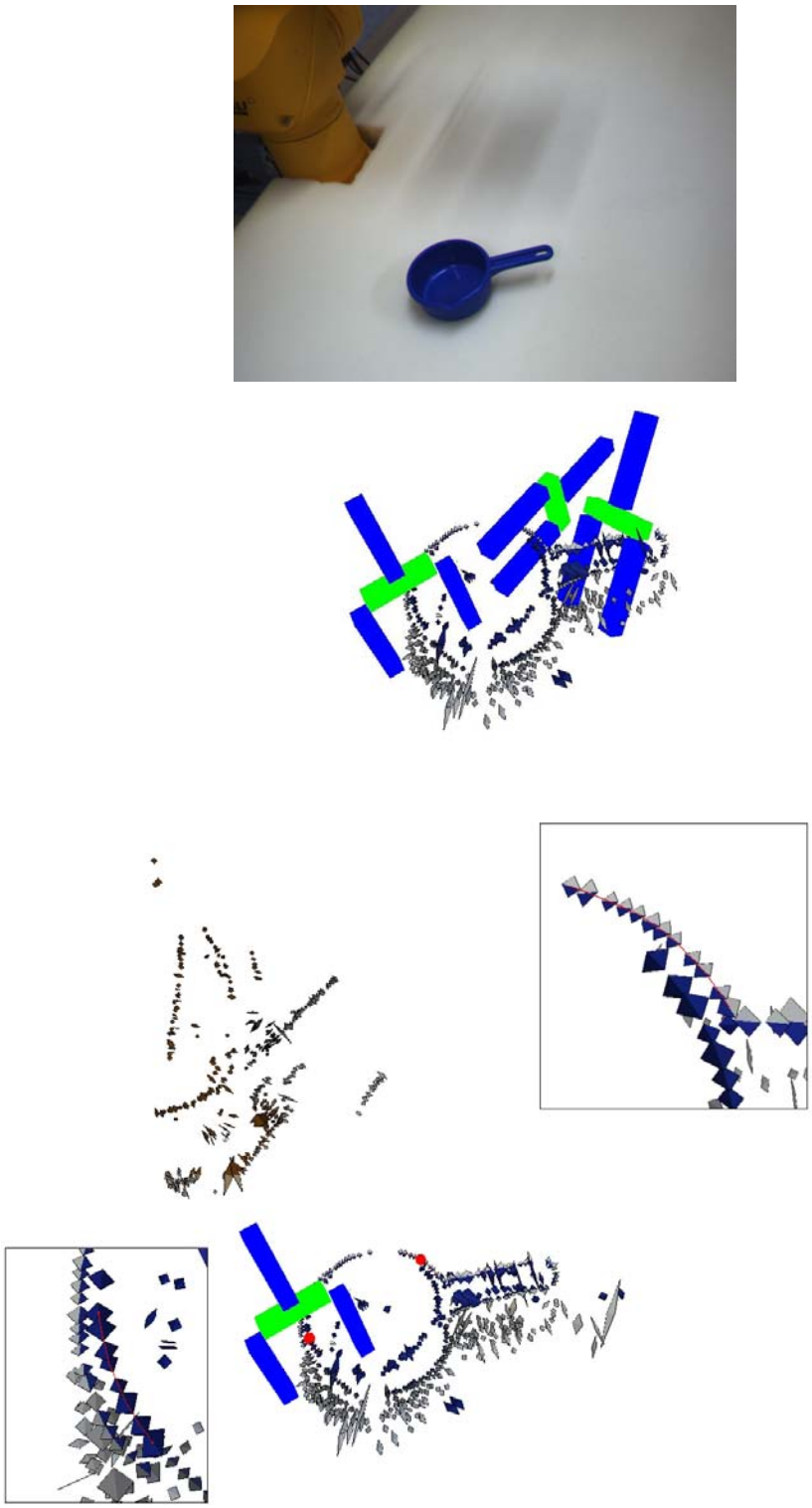


Figure 5.6: The top figure shows the image capture by the left camera taken during one of the experiments (Chapter 7.1). Middle and bottom figures are taken from WandererX visualisation environment. The middle image shows several grasping hypotheses. The bottom figure shows one grasping hypotheses that was successfully performed in experiment, together with the parent primitives and contours. The details of the parent contours are magnified.

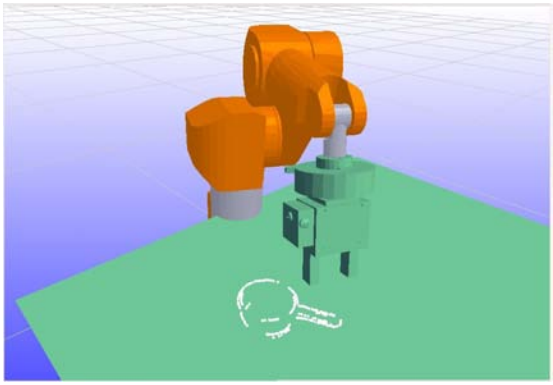


Figure 5.7: Robwork simulation environment shows 3D models of Staubli robot and floor. Additionally, the information about 3D edges in the scene is provided by the vision system, where the original scene is the same as on Figure 5.6. The 3D contours are composed of 3D primitives, which are modeled as small cubes.

few grasping hypotheses that will be tested before a new cycle begins. This is done by ranking derived grasping hypotheses based on different criteria.

Grasping hypotheses can be ranked based on position and orientation of the gripper. For example, if in a complex scene a grasp is positioned higher then others, there is a bigger chance that the object to be grasped is free and not held down by other objects, and that the approaching the object is easier. Another basis for ranking can be the confidence of 3D reconstruction [PKB+08], the amount of similarity (co-planarity and co-colourity) between parent contours, size of the parent contours or the amount of orthogonality to the connecting line between parent primitives (for EGA 1 and 2 types of grasps). Besides ranking, the choice of grasping hypotheses to be performed in one cycle can be based on grasps diversity, or can include some randomness.

As the grasping hypotheses set can contain several thousands elements, the system is not checking all of them for reachability and collision. In current implementation, the system instead starts by taking one by one grasps from top of the ranked set. If a grasp can be performed it is saved on the list of the grasps to be tested, othervise the algorithm proceeds to the next grasp on the list. When certain number of usable grasps is found the system proceeds to testing phase.

Individual grasp analysis and testing

Figure 5.8 shows how a grasp is performed. The gripper fingers are first set to the correct initial distance and the robot performs a movement from its initial “home” configuration to the “approach” configuration. In “approach”

configuration, the tool has the same orientation as in the grasping stage, and the final grasping configuration can be reached with a linear movement of the tool. After robot moves from the "approach" to the "grasp" configuration, the gripper closes (or opens) fingers to grasp an object. This is followed by "unapproach" movement of the robot, where tool is moved back to the approach position, and finally the robot moves to its initial configuration.

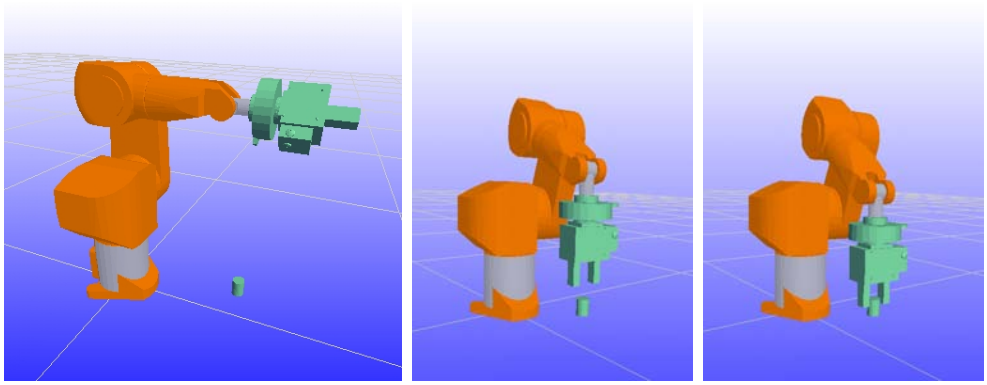


Figure 5.8: Left Figure shows "home" robot configuration - the default configuration robot has before and after performing a grasp. Middle and right Figures are examples of "approach" and "grasp" configurations.

A grasp can not be performed unless a collision free configurations can be found for both approach and grasp position of the tool. Both approach and grasp tool position/orientation can be reached with up to 8 different joint configurations. It is necessary to chose the optimal pair of approach-grasp configurations. The approach and grasp configuration have to match so that a linear movement of the tool between the two is possible.

The gripper has a (180°) symmetry around its z axis in respect to grasping, which means that additional configurations are available. These additional solutions are added to the solutions derived from inverse kinematics. This is important because gripper body's geometry is not symmetrical and some grasps that cause collision in original configuration could be collision free when the gripper is rotated 180° around its z axis.

If more than one pair of approach-grasp configurations are available, the system will give advantage to configurations that are closer to the "home" configuration. Since "home" is an "elbow up" configuration, (Figure 5.9), the "elbow up" configurations are favoured. Configurations closer to the "home" configuration are performed faster and are less prone to collision.

Grasp planning doesn't end when collision free robot configurations are found for approach and grasp case. The robot still has to move from the initial to the approach position and from the approach to the grasp position. During this movement the robot has to pass through number of different configurations,

Figure 5.9: Two different robot configurations lead to the same tool position. The left image shows "elbow down" and the right image "elbow up" configuration.

and it is necessary that those configurations are also collision free. The computation of the collision free paths is called motion planning. This work uses the RRT-connect (Rapidly-exploring Random Trees) motion planner [KL00].

The motion planner is used to calculate collision free path from "home" to "approach" position. If both "approach" and "grasp" robot configurations are collision free, the system assumes that the linear path between them is also collision free and this part of the path is not planned. The calculated "home - approach" path is also used when the robot moves in the opposite, "approach - home" direction, weather after performing a grasp or after a collision has been detected.

Although some collisions can be avoided by planning, they will still happen because the system is dealing with unknown scenes, and the motion planner is working with incomplete information. Apart from the robot and the floor models, the only information available to the planner are sparsly reconstructed contours that outline the objects on the scene. The surfaces are "invisible" for the planner. In order to protect the robot and environment, and in order to provide basis for autonomous exploration without human supervision, the system is equipped with force torque sensor that acts as an active collision detector. It is able to detect collision between the gripper and the surrounding and it is described in detail in Chapter 6.

The force torque collision detector is active during robot's home-approach and approach-grasp movements. Due to the nature of the collision checking method, it is not possible to check for collision in other situations. However, this is usually sufficient. If a collision is detected, the robot will stop and move backwards using the same path as for forward movement, until it reaches "home" position. The system will then proceed with testing next grasping hypotheses if any, or start a new vision processing cycle.

Grasp attempt outcome

The system is able to detect four different outcomes of testing a grasping hypotheses. This is done by monitoring the distance between gripper fingers.

If gripper fingers had to be closed in order to grasp an object (EGA 1, 3 and 4), then a **successful** grasp can be detected if final distance between fingers, when the robot has returned to its initial position, is bigger then zero. If gripper was opened to grasp an object (EGA 2), then a successful grasp is detected when final finger distance is smaller than maximal distance. It follows that zero final distance between fingers for EGA 1, 3 and 4, and maximal final distance between fingers for EGA 2 indicates an **unsuccessful** grasp. If distance between fingers indicated a successful grasp just after grasping and indicates unsuccessful grasp when robot reaches “home” position, then a grasp is characterised as **unstable**. The two movements - unapproaching" and "going home", that follow the grasping attempt, lift the object and serve as a primitive test of the quality of the grasp. The fourth outcome of the grasping attempt is **collision detected**. From a general point of view, unsuccessful, unstable and collision outcomes all fall under unsuccessful category.

In cases of unsuccessful or collision outcomes, the system continues testing remaining scheduled grasps if any, or starts a new image processing cycle otherwise. In case of a success the robot moves the tool to a position above a storing box and releases the object. Since successful grasps always, and unstable almost always change the scene, the system after performing them cancels any remaining grasp tests and proceeds directly to a new image processing cycle.

Chapter 6

Force Torque sensor

This chapter describes the FTACL 50-80 Schunk force torque sensor and explains how it is used for active collision detection during grasping procedure.

6.1 FTACL 50-80

The FTACL 50-80 is a combination of mechanical flexibility (springs displacement) and an optoelectronic position measurement system for all six degrees of freedom. It measures the full six components of force and torque and outputs the measured values in SI units in a 1kHz cycle. It doesn't require any calibration procedure. In addition it measures total (static and dynamic) acceleration and provides information of sensor displacement during operation [Sch]. It operates in temperature range of $5 - 55$ [℃] and supports standard interfaces (CAN, DeviceNET, RS232 and RS485). The sensor is shown on Figure 6.1 and overload limits are given in Table 6.1.

Figure 6.1: FTACL 50-80 Force Torque sensor manufactured by Schunk. Diameter of the sensor is 164 [mm].

F_x, F_y, F_z	300 [N]
T_x, T_y	7 [Nm]
T_z	15 [Nm]

Table 6.1: Operating limits for FTACL 50-80 sensor. Forces or torques out of permitted limits can permanently damage the sensor.

6.2 Active collision detection using the force torque sensor

In this project the Force Torque sensor is used for active collision detection. The sensor is mounted between the wrist and the tool of the robot (Figure 3.1), and it measures forces or torques acting on the tool. In order to detect collision it is necessary to separate the force and torque originating from gravitational force acting on the dead load, (empty gripper, upper part of the sensor and mechanical adapter connecting the gripper and the sensor), from any external contact force. In other words, if the difference between current measured force and torque values and the predicted values originating from gravitation is within certain limits, then it can be concluded that no forces apart from gravity are acting on the tool. Although the sensor provides information about acceleration forces, as well as information about small offsets to position and the orientation of the tool that arise from measurement itself, the satisfactory results were obtained without including the mentioned corrections into calculus.

$\mathbf{F}_m(F_{xm}, F_{ym}, F_{zm})$ and $\mathbf{T}_m(T_{xm}, T_{ym}, T_{zm})$ mark the measured output from force torque sensor, and similarly the $\mathbf{F}_g$ and $\mathbf{T}_g$ mark the effect of the gravity on the dead load. $\mathbf{F}_d$ and $\mathbf{T}_d$ are the differences between the two:

$$ {}^S\mathbf{F}_d = {}^S\mathbf{F}_m - {}^S\mathbf{F}_g, \tag{6.1} $$

$$ {}^S\mathbf{T}_d = {}^S\mathbf{T}_m - {}^S\mathbf{T}_g \tag{6.2} $$

$$ collision = \begin{cases} true & \text{if } \| \mathbf{F}_d \| > L_{force} \ \bigvee \ \| \mathbf{T}_d \| > L_{torque} \\ false & \text{otherwise} \end{cases} \tag{6.3} $$

where L_{force} and L_{torque} are the total force and the total torque collision limits. The index S indicates that all calculations and measurements are in the reference frame of the sensor S.

According to [Cra89], the following equations are applied for transforming force and torque from one coordinate system to another:

$$^{A}\mathbf{F} = {}^{A}R_{B} \cdot {}^{B}\mathbf{F} \tag{6.4}$$

$$^{A}\mathbf{T} = {}^{A}P_{Borg} \times {}^{A}R_{B} \cdot {}^{B}\mathbf{F} + {}^{A}R_{B} \cdot {}^{B}\mathbf{T} \tag{6.5}$$

where A and B are the two coordinate systems, and ${}^{A}P_{Borg}$ is the position of the origin of coordinate system B expressed in A.

In Equations 6.1 and 6.2, ${}^{S}\mathbf{F}_{m}$ and ${}^{S}\mathbf{T}_{m}$ are known from measurement and ${}^{S}\mathbf{F}_{g}$ and ${}^{S}\mathbf{T}_{g}$ have to be predicted for each tool orientation. The World coordinate system has been defined so that direction of the gravitational force is along negative Z axis. The gravitational force that acts on the dead load expressed in the coordinate system of its centre of mass is:

$$^{CM}\mathbf{F}_{g} = {}^{CM}R_{W} \cdot {}^{W}\mathbf{F}_{g} \tag{6.6}$$

$$^{W}\mathbf{F}_{g} = f_{g} \cdot (0, 0, -1)^{T}$$

where f_g is the weight of the dead load, index CM the Centre of mass reference frame, index W the World reference frame and ${}^{CM}R_{W}$ symbolises the rotation of the Centre of mass reference frame in respect to the World reference frame.

Using Equations 6.4 and 6.5 the gravitational force and torque can be expressed in the reference frame of the sensor. Having in mind that there is no rotation between S and CM reference frames and that no torque is present in the CM reference frame, following expressions are derived:

$$^{S}\mathbf{F}_{g} = {}^{CM}\mathbf{F}_{g} = {}^{S}R_{W} \cdot {}^{W}\mathbf{F}_{g} \tag{6.7}$$

$$^{S}\mathbf{T}_{g} = {}^{S}P_{CMorg} \times {}^{CM}\mathbf{F}_{g} + I \cdot {}^{CM}\mathbf{T}_{g} = {}^{S}P_{CMorg} \times {}^{S}\mathbf{F}_{g} \tag{6.8}$$

Rotation ${}^{S}R_{W}$ changes with time, as manipulator moves, and is available from forward kinematics of the robot when its configuration is known. ${}^{S}P_{CMorg}$ is the position of the centre of mass of the dead load expressed in the Sensor reference frame. Position of the Centre of mass and the weight of the dead load are the two unknown values. They are measured using following calibration procedure adopted from [Kra06].

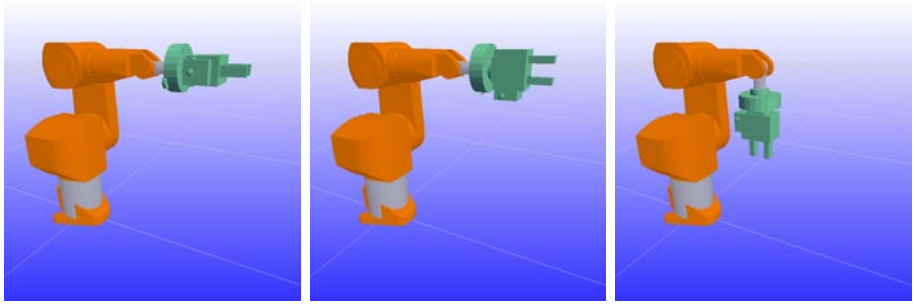


Figure 6.2: Three tool positions used for the calibration procedure.

The empty gripper is positioned into three different stable orientations (Figure 6.2) and the output of the sensor is recorded. For each orientation several measurements are made to minimise the influence of noise. The three orientations were chosen to capture the position of the centre of mass in all three dimensions. The weight of dead load is extracted in the following way:

$$f_g = \frac{1}{3}\sum_{i=0}^{3}\sqrt{f_{ix}^2 + f_{iy}^2 + f_{iz}^2} \tag{6.9}$$

Calculation of the centre of mass begins with Equation 6.8. The cross product is then rewritten in terms of matrix multiplication:

$$^{S}\mathbf{T}_g = {}^{S}F_g \cdot {}^{S}P_{CMorg} \tag{6.10}$$

$$^{S}F_g = \begin{pmatrix} 0 & f_{zg} & -f_{yg} \\ -f_{zg} & 0 & f_{xg} \\ f_{yg} & -f_{xg} & 0 \end{pmatrix}$$

Since three calibration orientations are used, Equation 6.10 is overdetermined. It is solved with least squares solution, that tries to minimises the Euclidean norm of the residual $\|{}^{S}F_g \cdot {}^{S}P_{CMorg} - {}^{S}\mathbf{T}_g\|$, [MWR].

6.3 Collision Limits

The collision limit values (Equation 6.3) were attained through experiments. The aim was to find the values that would be sensitive to collision, but high enough not to react to the acceleration, the deacceleration and the noise. Following values were found to be optimal:

$$L_{force} = 15N$$
$$L_{torque} = 0.5Nm$$

Figure 6.3 shows the differences between measured and calculated total force and total torque as a function of time for a sample grasp attempt that resulted in a collision. During first three seconds the robot is not moving and the orientation of the tool is constant. A small peek in the total force difference is detected in the moment when movement starts. It is caused by the acceleration and it stays within collision limits. The second peek, visible on both graphs, coresponds to the detected collision with an object. The shape of the graph in the area where collision happens indicates that it happens during the linear movement of the tool from the approach to the grasp position.

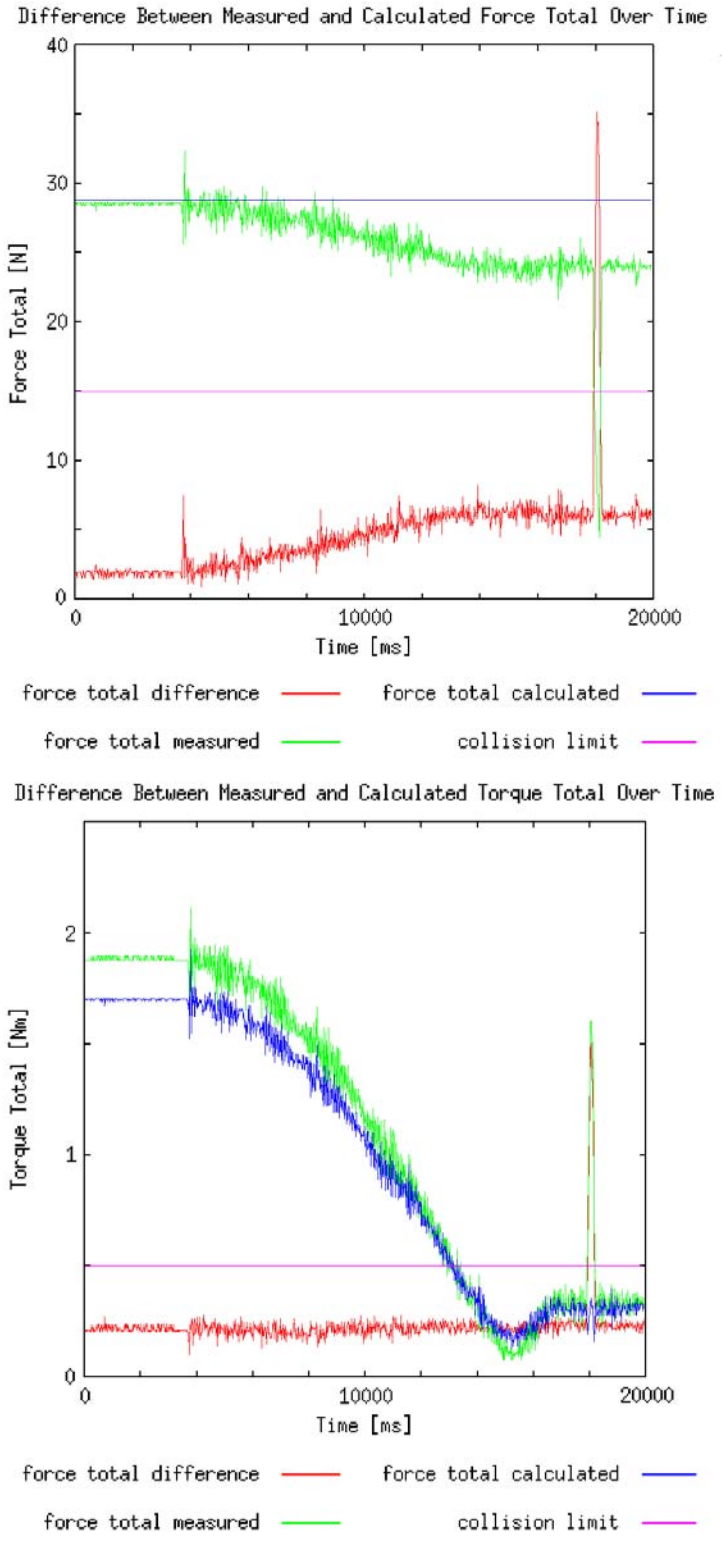


Figure 6.3: The graph shows the total measured and the total calculated forces (top) and torques (bottom), and the differences between measured and calculated values as a function of time for a sample grasping attempt where a collision happened.

Chapter 7

Evaluation

The evaluation presented in this work is designed as an exploratory case analysis. It is a first experimental evaluation carried out for this system. Since grasping reflex is a novel and complex procedure, any testing at this stage gives new insights and improvements are made daily. A systematic quantitative analysis would therefore be premature. This evaluation is designed to illustrate different aspects of system's behaviour, its capabilities and weaknesses. It is divided into two sections.

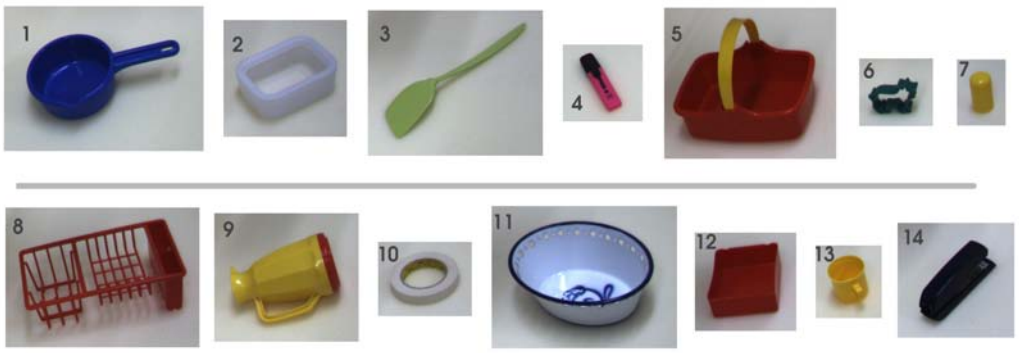


Figure 7.1: Office and toy kitchen objects used in evaluation. Objects are of mostly uniform colours, and their size and the shape is suitable for grasping with the parallel jaw gripper.

In the first Section 7.1, a test scene contains a single object and the robot attempts to remove it from the scene using the grasping reflex. Fourteen objects used in the evaluation are shown on Figure 7.1. The size and the shape of the objects is chosen so that grasping is possible with different difficulty. Objects are of uniform colours.

Second group of experiments are performed on five complex scenes containing the same objects, Section 7.2. For each scene two experiments are made. In

first the robot is programmed to perform random grasping actions, with no guidance from vision system, and in second the robot is using the grasping reflex. The goal is to remove as many objects as possible from the scene.

In final Section 7.3 the results of the experiments are discussed.

Exploration parameters

It is possible to make some adjustments to the flow of the exploration procedure with several paremeters. As explained in Chapter 5.2, grasping hypotheses within one set can be ranked according to different criteria. In this evaluation adopted ranking criteria is the amount of the verticality of the grasp, or more precisely:

$$R_S = Z_{TCP} \cdot -Z_W$$

where ranking score R_S is in interval [-1,1], Z_{TCP} is the orientation of the Z axis of the TCP frame expressed in the World reference frame, and Z_W is the $(0, 0, -1)^T$ vector. The grasps where the gripper fingers are pointing vertically down have the highest rank.

The maximal number of grasping attempts that is performed in one exploration cycle is 5. Grasp are chosen as described in Chapter 5.2, i.e. as first five best ranked grasps where both approach and grasp configurations are collision free, and a collision free path between home and approach configuration exists. Many grasping hypotheses will be discarded because of collision with the floor before they are performed. In order to attain more grasping possibilities, the the position of the floor surface has been lowered by 1 cm in the 3D model of the World. The system relies on the force torque sensor active collision detection to prevent damages. The TCP reference frame is positioned 18 millimetres from the finger ends, which means that grasps are 18 millimetres deep.

The parameters of the vision system used for extracting image representation are standard parameters, and the minimum number of primitives in a 3D contour is 3. The area of interest for parent primitives is a bounding box $X[250, 1000]$, $Y[-1000, 1000]$ and $Z[-228, 500]$ (millimetres).

7.1 Simple scenes - grasping reflex

Each of the fourteen objects has been presented to the system in several different positions and orientations. Experiments performed with the first object are described with most detail. Other experiments are given briefly unless they illustrate an aspect that has not jet been described.

Object 1

Three experiments were performed with object 1, (Figure 7.2). In the first one the object was successfully grasped in the first attempt with grasp of EGA 3 type. Figure 7.3 shows the good grasping hypotheses in WandererX visualisation environment together with the other three grasping hypotheses generated by the same two contours. One contour originates from the handle and other one from the body of the object.

In the second experiment the first grasping attempt, of EGA 3 type, again gave good results. This time both edges were originating from the body of the object (Figure 5.6). The number of generated grasping hypotheses, together with the number of processed, good, unreachable, those grasping hypotheses that cause collision and those where the motion planner didn't find solution is listed in Table 7.1.

Figure 7.2: Three experimental situations for object 1. Figure shows original images used for acquiring image representations, captured by left camera. The darker area in the middle of all three images is a shadow robot makes when in initial position.

	1	2	3a	3b	3c
number of grasping hypotheses (GH)	66	373	45	55	47
number of processed GHs	12	11	45	55	47
number of accepted GHs	5	5	0	3	0
number of unreachable GHs	7	1	44	45	40
number of GHs in collision	0	5	1	7	7
number of GHs where no path was found	0	0	0	0	0

Table 7.1: The results of processing grasping hypotheses (GHs) in order to find those that can be performed. Columns 1, 2 and 3 stand for the three experiments. In the third experiment the system went through three exploration cycles (a, b, c).

The ranked list of grasping hypotheses (GHs) is processed top-down. The processing stops when certain number (5 here) of accessible GHs has been

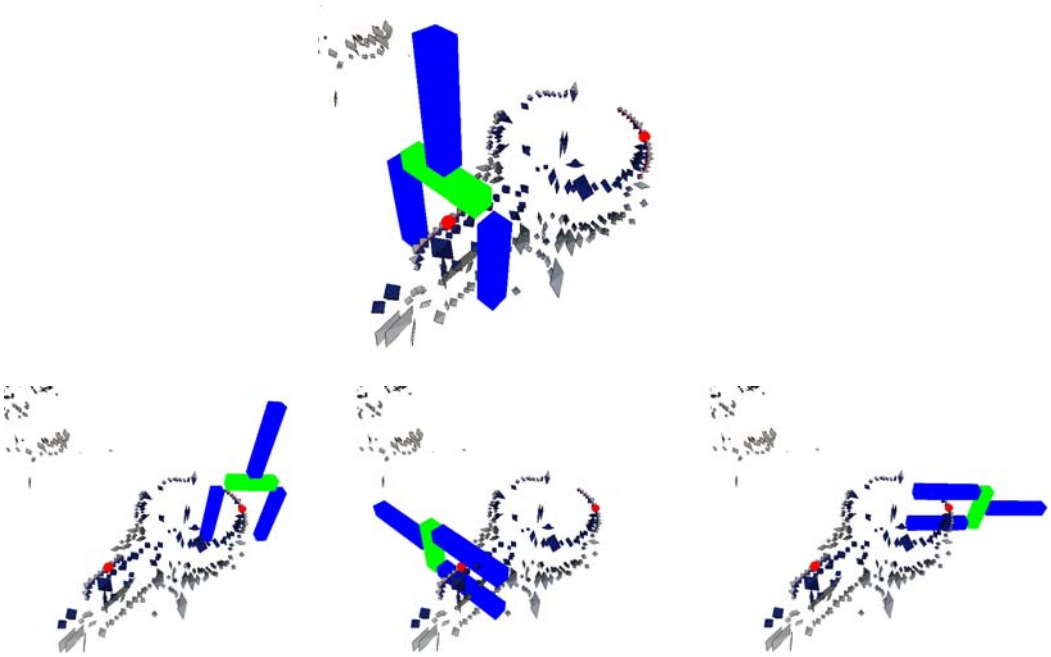


Figure 7.3: The top figure shows the grasping hypothesis of type EGA 3 that resulted in successful grasping. Parent primitives are also visible. The bottom images show the other three grasping hypotheses (one EGA 3 and two EGA 4 grasps) generated by the same parent primitives pair.

found, or when there are no more GHs available (Chapter 5.2). In order to give a better illustration of the processing outcome, full sets of GHs has been processed for first and second experimental situation, and the results are shown in Table 7.2.

	1	2
number of grasping hypotheses (GH)	66	373
number of processed GHs	66	373
number of accepted GHs	11	37
number of unreachable GHs	46	243
number of GHs in collision	9	93
number of GHs where no path was found	0	0

Table 7.2: The results of processing full sets of grasping hypotheses for first and second experimental situation.

Table 7.3 shows some intermidiate values from the grasping hypotheses generation program for the same two experiments. Number of contours and similar contour pairs is derived from the whole image representation. Parent primitive pairs are then assigned. A parent pair is discarded if any of the two primitives doesn't belong to a certain region of interest. Background features that originate from the robot and the edge of the ground surface (Figure 7.2) generate a lot of undesirable similar contours and that is why the number of discarded parent pairs is high. This however doesn't explain why there is a significant difference between number of good parent pairs and consequently generated grasping hypotheses in the two cases. This difference arises because the representation of the Object 1 contains less detail in the first case, Figure 7.4.

	1	2
number of contours	27	30
number of similar contours pairs	201	241
number of parents pairs	17	94
number of discarded parents	184	147
number of GHs	66	373

Table 7.3: The table presents some intermidiate values from grasping hypotheses generation program. The values in column 1 originate from the first experiment, and values in column 2 from the second.

In the third experiment no grasps were performed initially. Although 45 grasping hypotheses were generated, 44 of them were out of robot's reach and one was causing collision, Table 7.1. The system then repeated the exploration

Figure 7.4: Object 1 on two different image representations, taken from the first and the second experiment. The representation on the left contains somewhat less detail and gave a smaller number of GHs than the right representation.

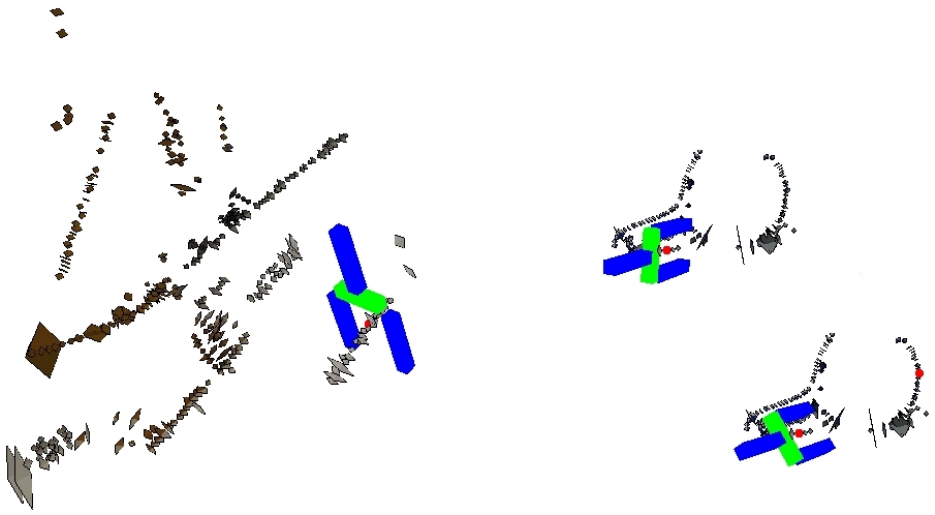


Figure 7.5: Three unsuccessful grasping hypotheses generated in the second cycle of the third experiment. The original scene is the rightmost image of Figure 7.2. All three hypotheses were generated by shadows.

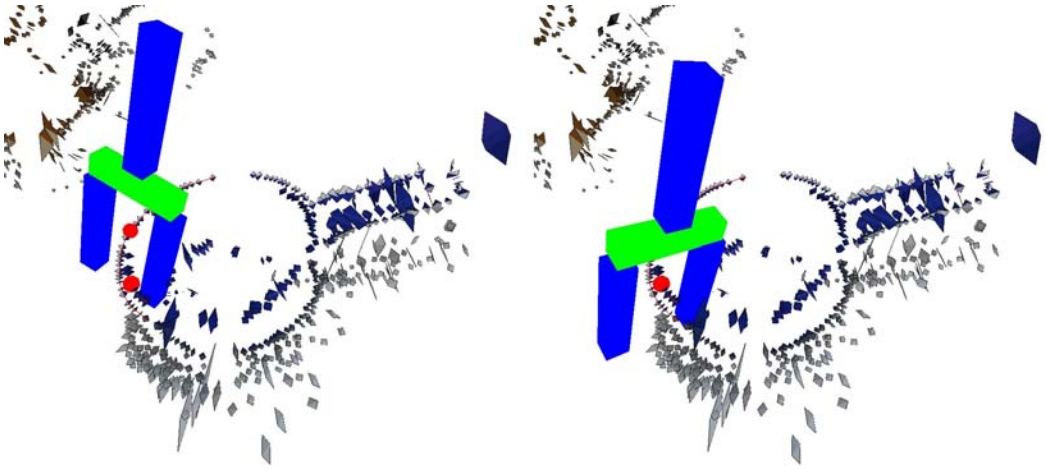


Figure 7.6: Two grasping hypotheses of EGA 3 type generated by concentric neighbour contours.

cycle, i.e. the cycle of capturing new images, producing new image representation and new set of grasping hypotheses, two more times before execution was stopped. It is interesting to notice that the three cycles gave slightly different image representations and therefore different sets of grasping hypotheses, while the scene and the lighting has not been visibly changed and the image capturing happened in several minutes intervals. In the second cycle, three grasping attempts were made, all three unsuccessful. Figure 7.5 reveals that the grasping hypotheses were triggered by contours originating from robot's and object's shadows. No grasping attempts were made in third cycle.

Table 7.2 shows that the motion planner always finds the collision free path between "home" and "approach" configurations. This is because the motion planning task is simple in most cases. The role of the motion planner is to find a collisionfree path for the robot approach movement that takes into account the 3D edges information provided by the vision system. In cases when resulting path is complicated the robot movements can seem unexpected. Figure 7.6 displays a pair of neighbour concentric similar contours with corresponding grasping hypotheses.

Object 2

Figure 7.7 shows testing situations for the second object. Outcomes of the experiments are given in Table 7.4. In the first experiment four grasping attempts resulted in collision before last one succeeded. Figure 7.8 shows second grasping attempt. The 3D primitive that is representing the contour is chosen as a primitive in the middle of the contour (Chapter 5). As the process of 3D reconstruction contains uncertainties, positions and orientations

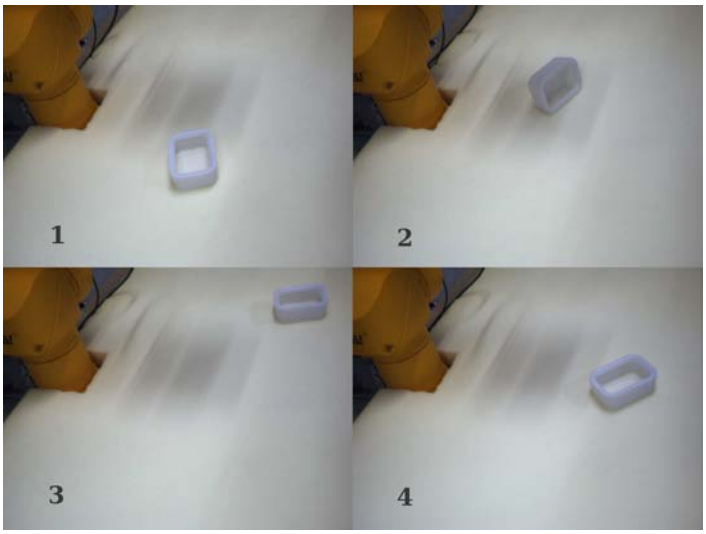


Figure 7.7: Test scenes for object 2.

of individual 3D primitives are not always reliable [PKB+08].

In the second testing situation, 9 out of 10 grasping attempts resulted in collision (7.7, 2). The grasping hypotheses are ranked by orientation of the tool, and in every new cycle of producing and choosing grasping hypotheses, the "vertical" grasps are chosen as first ones to perform. In this way the program is caught in a loop because vertical grasps were of EGA 3 and 4 type, assuming that the the concavity of the object is accessible from the top, and thus colliding with the top surface. An EGA 1 grasp acting on two top longer edges of the object could have given good results, but in this case only one of the two edges was reconstructable. The need for better ranking strategy, and better strategy for choosing which grasps to perform first, becomes apparent.

In the third experiment the reconstruction was very poor, because the object was far away (Figure 7.7, 3). The process of acquiring image representation was done two times. First cycle gave 8, and second 4 grasping hypotheses and only two grasping hypotheses could be performed. In contrast to third, fourth experiment gave excellent reconstruction. The two reconstructions are compared on Figure 7.9.

Object 3

The four test scenes for object 3 are shown on Figure 7.10. This object triggers grasps that are very close to the floor. The grasp can be successful only if the gripper is positioned very precisely to reach the object, but not touch the floor. While in this situation a more "shallow" position of TCP reference frame is preferred, in many other cases a "deeper" grasp is better because contact area

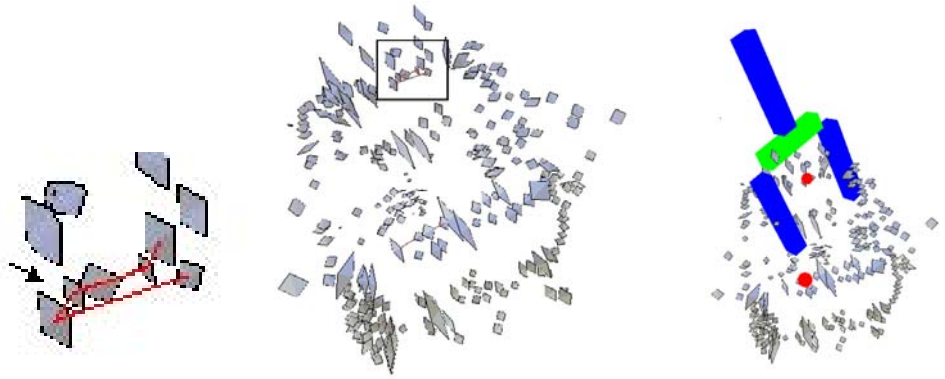


Figure 7.8: The grasping hypothesis that resulted in collision. The detail from the middle image is shown on the left. The primitives connected with the red line belong to one of the two 3D contours used for constructing the grasping hypothesis. The arrow points to the parent primitive that is representing the contour. The primitive's orientation deviates from the object's edge orientation, which is reflected on the resulting grasping hypotheses on the right image where gripper fingers are colliding with the edge of the object. The ordering of primitives in respect to their 3D position doesn't appear correct because of the chosen point of view.

	1	2	3	4
number of grasping attempts	5	10	2	1
number of exploration cycles	1	2	2	1
successful grasps	1	0	0	1
unstable grasps	0	0	0	0
collisions	4	9	0	0
unsuccessful grasps	0	1	2	0

Table 7.4: Experiments results for object 2.

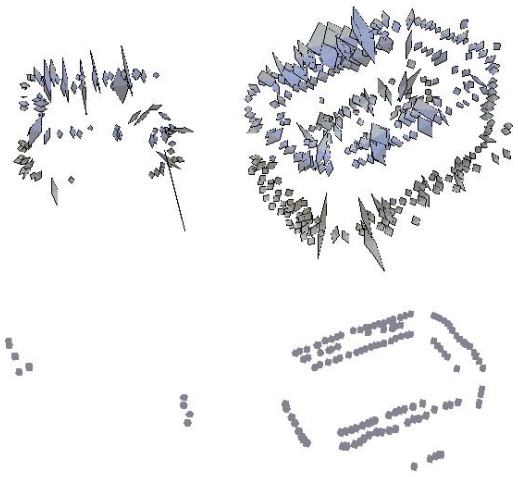


Figure 7.9: Top: Image representations of scene 3 and 4 (Figure 7.7) in WandererX visualisation environment. Bottom: corresponding 3D contours in Robwork environment, where contours are added to 3D model of the world. This model is used for motion planning (Chapter 5.2).

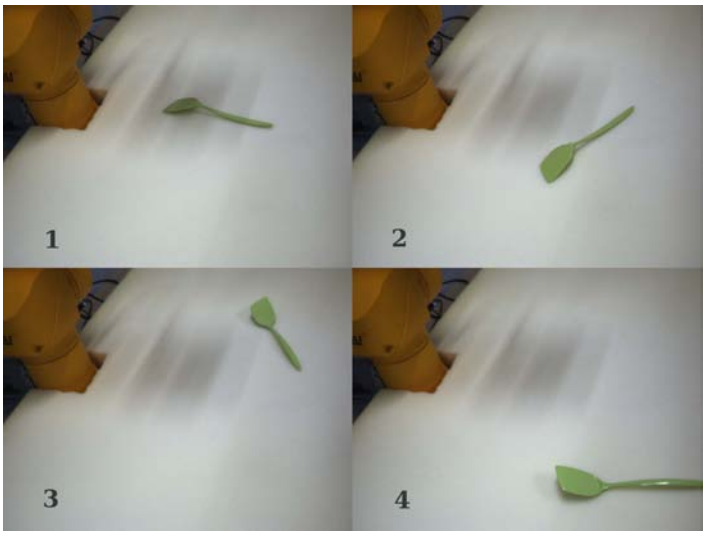


Figure 7.10: Test scenes for object 3.

between gripper and object is bigger and the grasp becomes more stable. The results of the experiments are given in Table 7.5. In the current system the depth of the grasp is a constant for all cases.

	1	2	3	4
number of grasping attempts	8	1	3	1
number of exploration cycles	4	2	3	3
successful grasps	0	1	0	1
unstable grasps	1	0	0	0
collisions	5	0	0	0
unsuccessful grasps	2	0	3	0

Table 7.5: Outcome of tests on object 3. In the second and fourth scene the object was successfully grasped with first grasping attempts. Those attempts were, however, generated after two cycles of capturing images for the second scene, and after three cycles for the fourth scene.

The five grasps that are performed in one exploration cycle can sometimes be too similar. The first and second grasping attempt for the first scene (7.10, 1) had following coordinates:

$$g_1 = (332.832, 13.7542, -215.936, 150.408, -4.35508, 95.1484)$$
$$g_2 = (332.832, 13.7542, -215.936, 148.199, -4.28776, 95.0478)$$

where the first three values are x, y and z coordinates of the position of the TCP frame in respect to the World frame in millimetres, and the last three values give XYZ Euler rotation of the TCP frame in degrees. This situation suggests that a method for choosing grasping hypotheses to be performed in one cycle should favour diversity.

Figure 7.11 shows a common situation where a grasping attempt moves an object. When this happens, the remaining grasping attempts of the same exploration cycle that are performed on the same object might not make sense. Because of its elasticity a foam floor often returns objects to its initial position, making the remaining grasping hypotheses still valid. On the other hand, if during grasping one gripper finger touches the object before the other, the pressure it applies on the object makes the object "slide" into optimal grasping position when hard floor surface is used. This is not the case with the foam surface where elasticity can be a problem. The grasping hypotheses on Figure 7.12 right, resulted in unstable grasp because of the floor resistance.

Figure 7.13 shows two successful grasping hypotheses. In both cases the object is grasped by features that are highest above the floor. Although object was grasped with one attempt in each case, this successfull grasping hypotheses has

Figure 7.11: The Figure shows how an object is moved after unsuccessful graping attempt. Following grasping attempts that are scheduled for execution before next exploration cycle are not any more valid.

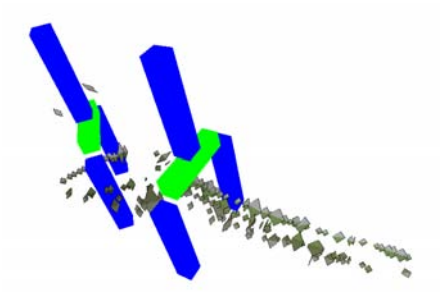


Figure 7.12: Two tested grasping hypotheses from the first scene on Figure 7.10. The grasping attempt shown on the left resulted in collision, and the grasp on the right was unstable.

been generated in respectively second and third exploration round. In other words, the first scene (scene 2 on Figure 7.10) did not give any acceptable grasping hypotheses in first exploration cycle and the second scene (scene 4 on Figure 7.10) did not give any grasping attempts in first two cycles. The minimal variation in lighting that occur each time images are captured can produce very different results. The second image on the figure shows very poor reconstruction of the handle of the object because its orientation is aligned with the epipolar line.

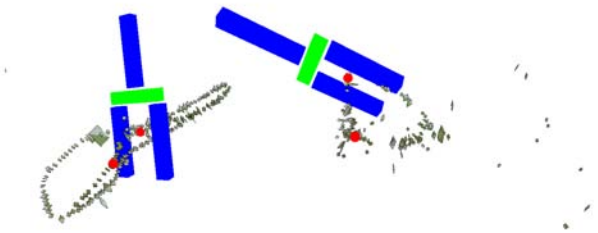


Figure 7.13: Two successful grasps from scenes 2 and 4 (Figure 7.10). They succeeded because the grasped features are slightly above the floor. The reconstruction of the object's handle on the scene 4 is especially poor because its horizontal orientation that matches epipolar line.

Object 4

Object 4 test scenes are shown on Figure 7.14. The results of experiments are given in Table 7.6. In the second experiment the object was successfully grasped by the black top.

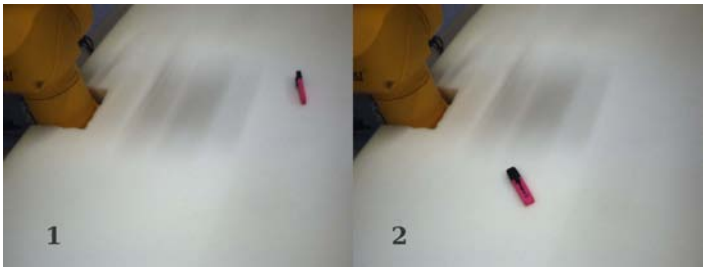


Figure 7.14: Test scenes for object 4.

	1	2
number of grasping attempts	0	1
number of exploration cycles	3	1
successful grasps	0	1
unstable grasps	0	0
collisions	0	0
unsuccessful grasps	0	0

Table 7.6: Experiments results for object 4.

Object 5

Table 7.7 gives results of experiments performed on object 7. The experimental situations are shown on Figure 7.15.

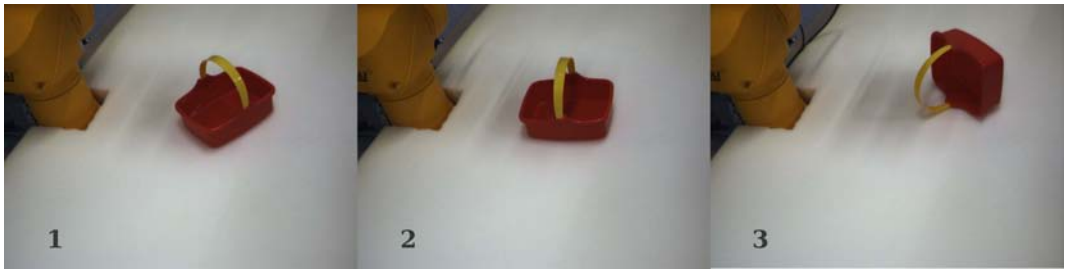


Figure 7.15: Test scenes for object 5.

	1	2	3
number of grasping attempts	1	3	6
number of exploration cycles	1	1	2
successful grasps	1	1	1
unstable grasps	0	0	0
collisions	0	2	3
unsuccessful grasps	0	0	2

Table 7.7: Experiments results for object 5.

Object 6

Object 6 test scenes are shown on Figure 7.16. Although grasp attempts often move an object, the scene 1 is the first case where resulting scene is explicitly shown (1a) because the majority of the explorations cycles (three) were performed with the new scene. The results of experiments are given in

Table 7.8. The image representation of object 6 usually contains only one contour and no grasping hypotheses can be generated. When the object is placed in a complex scene it can "borrow" the second contour from another object.

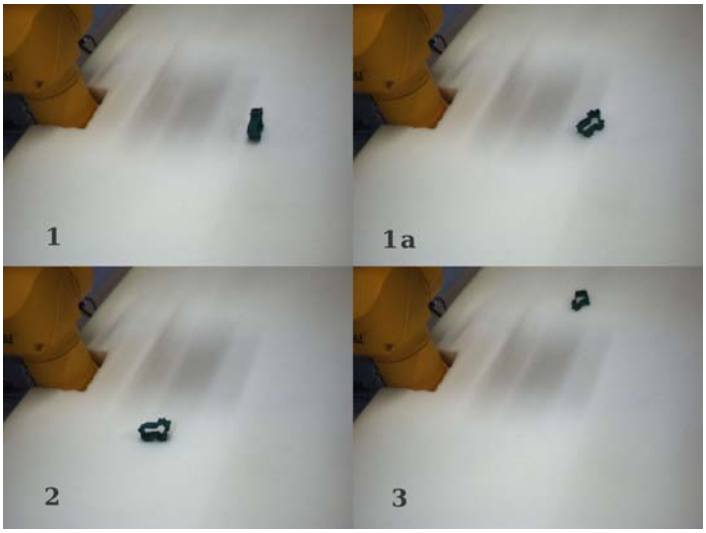


Figure 7.16: Test scenes for object 6. In the first exploration cycle of the first experiment (1) the object was moved by a grasping attempt to a new position (1a). The three remaining exploration cycles of the first experiment are performed on the scene 1a.

	1	2	3
number of grasping attempts	1	0	0
number of exploration cycles	4	1	1
successful grasps	0	0	0
unstable grasps	0	0	0
collisions	1	0	0
unsuccessful grasps	0	0	0

Table 7.8: Experiments results for object 6.

Object 7

Object 7 test scenes are shown on Figure 7.17. The results of experiments are given in Table 7.9.

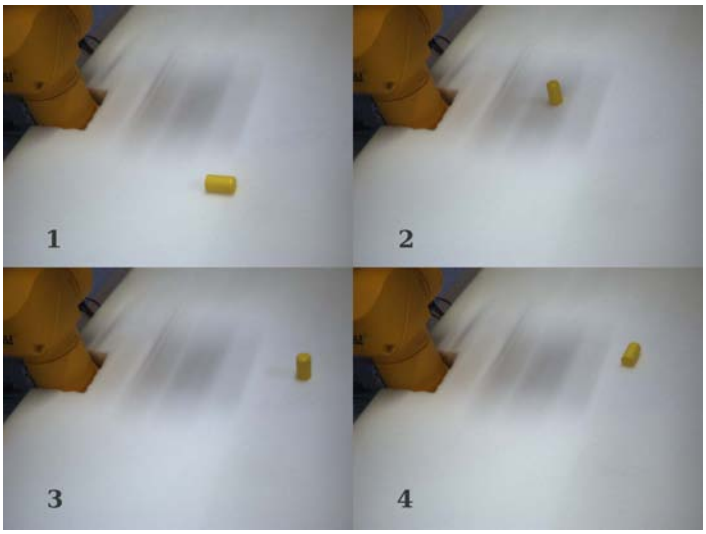


Figure 7.17: Test scenes for object 7.

	1	2	3	4
number of grasping attempts	3	0	1	0
number of exploration cycles	3	2	2	2
successful grasps	0	0	0	0
unstable grasps	0	0	0	0
collisions	2	0	0	0
unsuccessful grasps	1	0	1	0

Table 7.9: Experiments results for object 7.

Object 8

Table 7.10 gives results of experiments performed on object 8. The experimental situations are shown on Figure 7.18. In the first experiment a rare EGA 2 grasp succeeded. The grasping hypothesis is shown on Figure 7.19.

	1	2	3	4
number of grasping attempts	2	2	5	7
number of exploration cycles	1	1	4	3
successful grasps	1	1	0	0
unstable grasps	0	0	3	1
collisions	1	1	1	3
unsuccessful grasps	0	0	1	3

Table 7.10: Experiments results for object 8.

Figure 7.18: Test scenes for object 8.

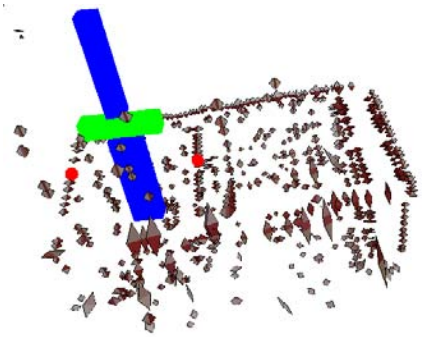


Figure 7.19: A successful grasping hypotheses of EGA 2 type from the first experiment (Figure 7.18 1). In EGA 2 type of grasp fingers are initially closed. The grasping is achieved by applying force from inside out.

Object 9

Object 9 test scenes are shown on Figure 7.20. The results of experiments are given in Table 7.11. It is difficult to grasp this object using the grasping reflex. The object has many edges, but few concavities and it is mostly too wide for EGA 1 type of grasp. The majority of the grasping hypotheses resulted in collision.

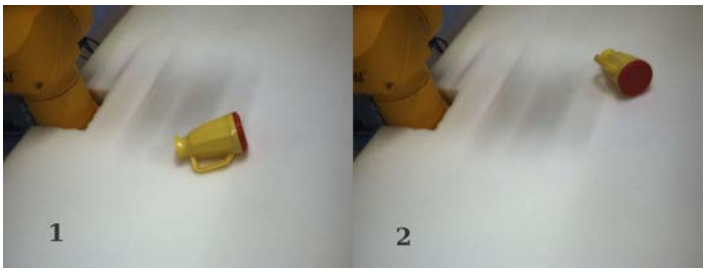


Figure 7.20: Test scenes for object 9.

	1	2
number of grasping attempts	5	11
number of exploration cycles	1	3
successful grasps	0	0
unstable grasps	0	0
collisions	5	6
unsuccessful grasps	0	5

Table 7.11: Experiments results for object 9.

Object 10

Table 7.12 gives results of experiments performed on object 10. The experimental situations are shown on Figure 7.21.

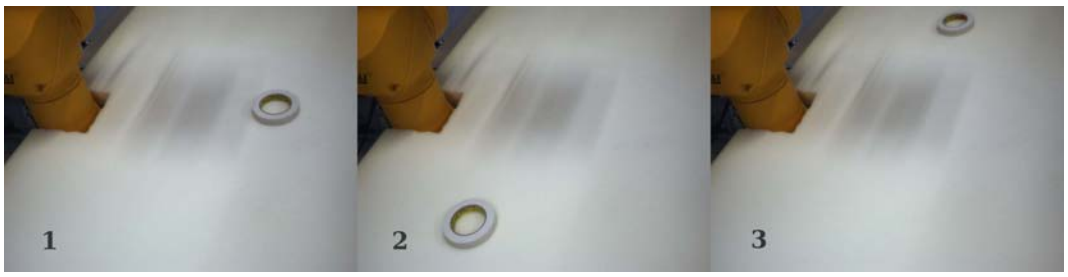


Figure 7.21: Test scenes for object 10.

Although the grasping hypothesis on Figure 7.22 seems correct, it did not succeed because the gripper did not reach the object and the fingers were closed above the object. The cause of this could be uncertainty of 3D reconstruction or inaccuracy in robot-camera calibration.

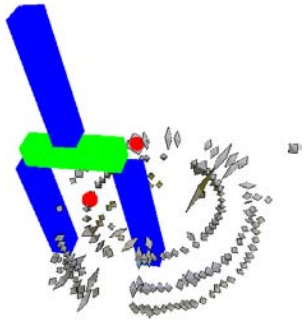


Figure 7.22: The figure shows a grasping hypotheses in wandererX visualisation environment. Although this grasping hypotheses seems reasonable, it did not succeed because the gripper did not reach the object before grasping the fingers.

	1	2	3
number of grasping attempts	1	2	3
number of exploration cycles	1	1	3
successful grasps	1	1	0
unstable grasps	0	0	0
collisions	0	0	1
unsuccessful grasps	0	1	2

Table 7.12: Experiments results for object 10.

Object 11

Object 11 test scene is shown on Figure 7.23. Result of experiment is given in Table 7.13. The system easily produces good grasping hypotheses for this object, but the force applied by the gripper is not enough to lift the heavy object, so the object is dropped.

Figure 7.23: Test scene for object 11.

	1
number of grasping attempts	1
number of exploration cycles	1
successful grasps	0
unstable grasps	1
collisions	0
unsuccessful grasps	0

Table 7.13: Experiments results for object 11.

Object 12

Table 7.14 gives results of experiments performed on object 12. The experimental situations are shown on Figure 7.24. The simple shape of this object makes it easy to grasp. Unfortunately two of four edges that can be grasped are too thin to be detectable with the gripper. Successful grasps of this object are sometimes characterised as unsuccessful and sometimes as unstable.

The simple shape of the object also means that few grasping hypotheses are made. That is why processing of vertically ranked grasping hypotheses have reached a horizontal grasp in the third experiment. The performed horizontal grasp of EGA 3 type (Figure 7.25 left) succeeded initially, but was characterised as unstable.

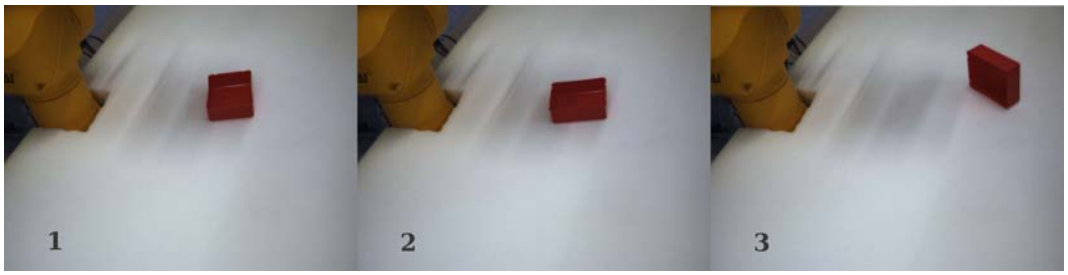


Figure 7.24: Test scenes for object 12.

Object 13

Object 13 test scene is shown on Figure 7.26. Result of experiment is given in Table 7.15. Grasps of EGA 1 type are rarely performed because parent primitives have to be similar (co-planar and co-colour), orthogonal to the line connecting them, and in the same time at most 58 mm away from each other. A wrong, but very precise grasping hypotheses of type EGA 4 (Figure 7.27 left) was observed in the third experiment. Examination of the grasping hypotheses

	1	2	3
number of grasping attempts	1	1	4
number of exploration cycles	1	1	2
successful grasps	0	1	0
unstable grasps	0	0	2
collisions	0	0	0
unsuccessful grasps	1	0	2

Table 7.14: Experiments results for object 12.

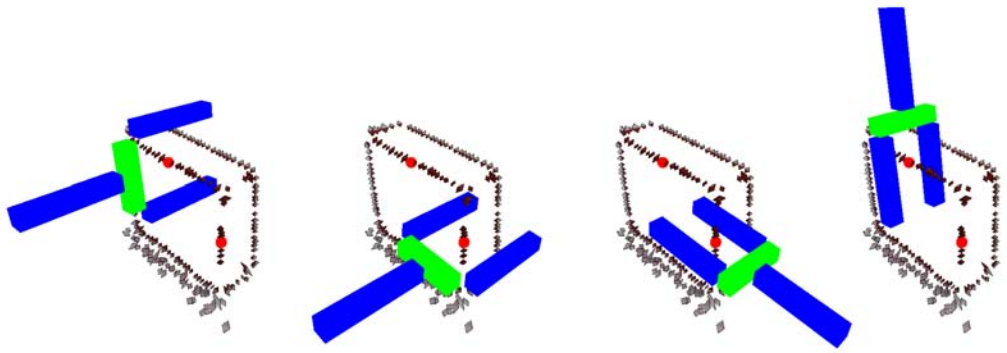


Figure 7.25: The grasping hypotheses on the left was successful (object 12, scene 3), but was registered as unstable because grasped edge is too thin and gripper fingers had a zero distance while grasping. The remaining three images show other grasping hypotheses generated from the same parent contours and illustrate a good reconstruction. This figure illustrates well why EGA 1 and EGA 2 types of grasps demand orthogonality of the parent primitives to the line connecting them. Neither EGA 1 nor EGA 2 type of grasp would give a stable grasp in this situation.

set in the WandererX visualisation environment revealed that corresponding EGA 1 was created (Figure 7.27 right). It was not performed because the approach configuration was not reachable by the robot. In the fourth experiment a successful grasp of EGA 1 type was carried out, (Figure 7.28).

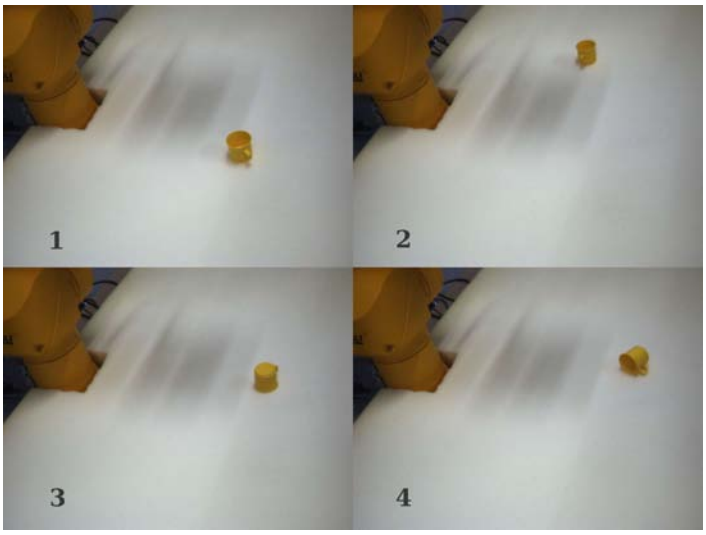


Figure 7.26: Test scenes for object 13.

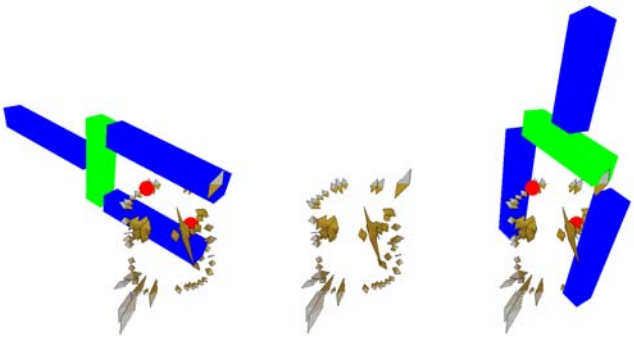


Figure 7.27: The wrong EGA 4 hypotheses (left) was performed, whereas the correct grasping hypotheses of type EGA 1 (right) was not because the approach configuration was unreachable by the robot.

	1	2	3	4
number of grasping attempts	2	0	2	6
number of exploration cycles	1	1	3	6
successful grasps	1	0	0	1
unstable grasps	0	0	0	0
collisions	1	0	0	4
unsuccessful grasps	0	0	2	1

Table 7.15: Experiments results for object 13.

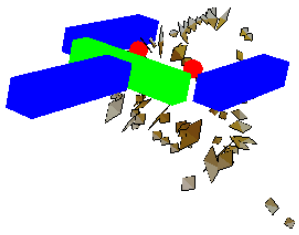


Figure 7.28: A successful EGA 1 grasp from the fourth experiment, Figure 7.26, 4.

Object 14

Figure 7.29: Test scenes for object 14.

	1	2	3
number of grasping attempts	6	3	8
number of exploration cycles	3	2	2
successful grasps	1	0	0
unstable grasps	2	0	0
collisions	3	3	8
unsuccessful grasps	0	0	0

Table 7.16: Experiments results for object 14.

7.2 Complex scenes - random grasping and grasping reflex

The second group of experiments is performed on five complex scenes. For each scene two kind of experiments are performed. In first experiment a 30 random grasping attempts is made. Grasps have a random orientation and random position inside area of interest inside the bounding box with following edges X[200, 700], Y[-700, 700], Z[-300, 0] (mm) and are not ranked in any way. The type of grasp (opening or closing fingers) is also random, but majority of grasps is performed by closing fingers. The 30 grasps are chosen from a larger set so that they fulfil the same reachability and collisionfree requirements as in the case of grasping reflex.

The second experiment for each scene is repeated on the identical scene (scene was reconstructed), but using the grasping reflex. The results of the random grasping and grasping reflex are roughly compared. The relative success of the grasping reflex off course depend on the number of the attempts taken into account. The 30 grasping attempts are usually enough for the system to

perform all possible successful grasps. If the system continues working after this point, the number of the unsuccessful, collision and unstable outcomes grows.

Figure 7.30: Complex scene 1. Left image shows the initial scene for both random grasping and grasping reflex experiments. The image on the right shows the scene after 30 grasping attempts with grasping reflex.

	random	grasping reflex
number of grasping attempts	30	30
successful grasps	0	6
unstable grasps	0	5
collisions	3	18
unsuccessful grasps	27	1

Table 7.17: The results of experiments with complex scene 1.

The first complex scene is shown on Figure 7.30, left. The corresponding outcomes of experiments are given in Table 7.17. In a complex scene grasping hypotheses can be defined with edges from two different objects. The definition of co-colourity (Chapter 4) says that two primitives are co-colour if their parts that face each other have the same colour. The outer colour of edges of the two objects is usually the colour of the floor surface and if the two edges are co-planar in the same time, a grasping hypotheses will be created. In most cases this is an advantage compared to single object scenes.

When an object is successfully grasped, it becomes attached to the robot's body and the path robot used for approaching the object might not any more be collision free. In that case, the attached object will either push the objects on its way, or the grasp will fail. A way to solve this is to give advantage to the grasps with higher position as they are less likely to be held down by other objects.

The right image on Figure 7.30 shows how the scene looked after 30 grasping reflex actions. Most of the remaining graspable objects are out of reach. The object 12 is still in reach, but is not being grasped. The system is repeatedly performing grasps on the object 11 with collision outcomes. The blue contours of that object are nicely reconstructed, and because of their horizontal position they trigger the vertical grasps that are performed in front of any others. This one more time indicates that vertical orientation alone is maybe not the optimal ranking criteria and that the choice of grasping hypotheses to be performed in one exploration cycle should include grasps diversity.

The results of the experiments on the second complex scene are shown in Table 7.18. The initial scene is shown on Figure 7.31 left, and the final scene is shown on the right. Remaining objects are mostly in tough positions, where concave features are not accessible. Only two objects were successfully grasped. The system often performed an unstable grasp of the object 1 by the handle, which failed because the object was held down by object 11.

Figure 7.31: Complex scene 2. Left image shows the initial scene for both random grasping and grasping reflex experiments. The image on the right shows the scene after 30 grasping attempts with grasping reflex.

	random	grasping reflex
number of grasping attempts	30	30
successful grasps	0	2
unstable grasps	1	5
collisions	4	13
unsuccessful grasps	26	10

Table 7.18: The results of experiments with complex scene 2.

The first grasping reflex in the third complex scene successfully removed two objects (12 and 7) from the scene. The initial scene is visible on Figure 7.32,

top left. The bottom image shows the successful grasping hypothesis. The grasping hypothesis originates from the objects 7 and 13, and the simultaneous grasping of object 12 together with object 7 was a coincidence.

	random	grasping reflex
number of grasping attempts	30	30
successful grasps	0	4
unstable grasps	0	2
collisions	4	12
unsuccessful grasps	26	12

Table 7.19: The results of experiments with complex scene 3.

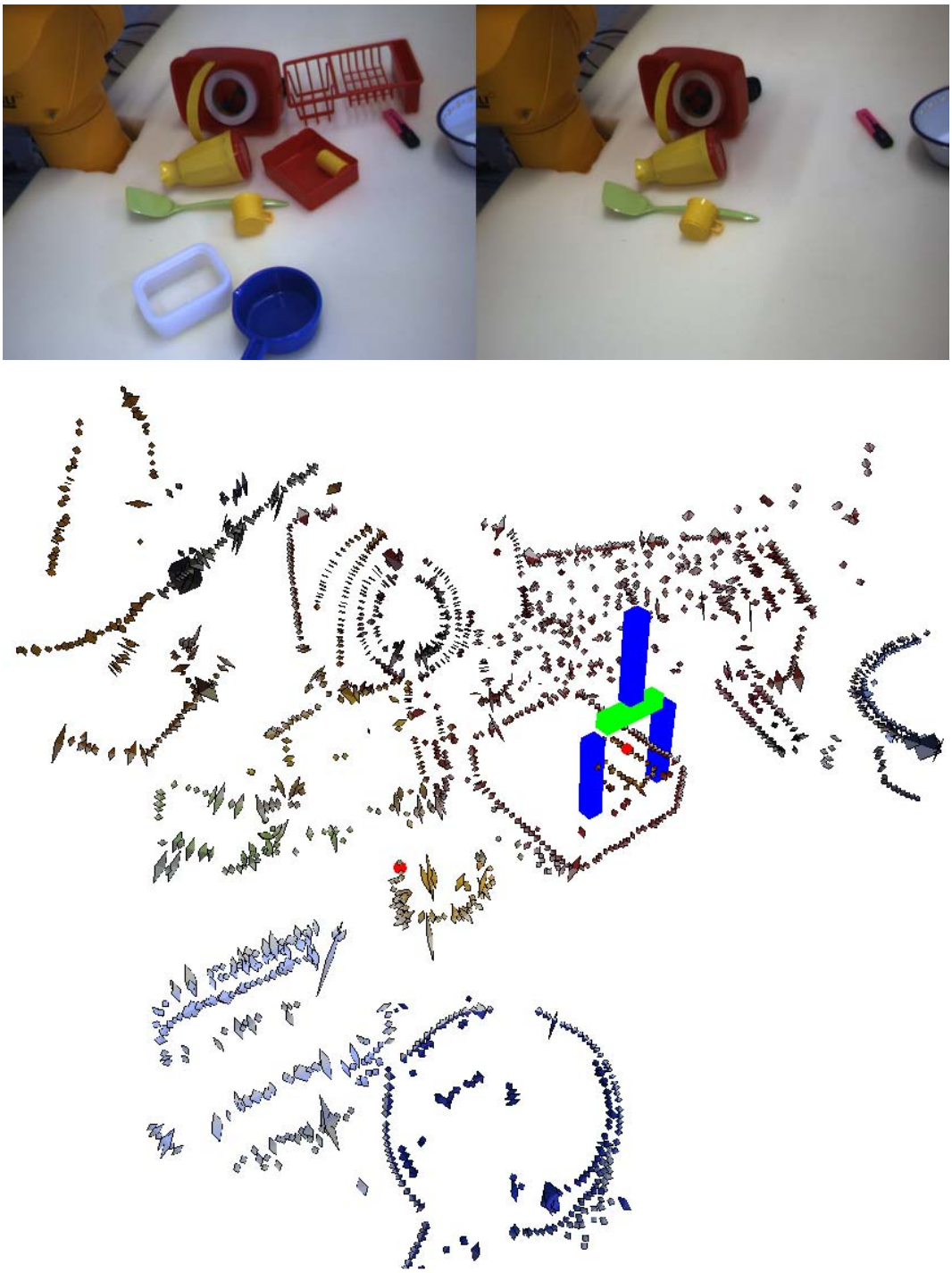


Figure 7.32: Complex scene 3. Top: Left image shows the initial scene for both random grasping and grasping reflex experiments. The image on the right shows the scene after 30 grasping attempts with grasping reflex. Bottom: the first grasping reflex attempt resulted in simultaneous grasping of two objects. Although objects 12 and 7 were grasped, the parent primitives (red dots) indicate that the grasping hypothesis originated from objects 7 and 13.

Figure 7.33 shows the fourth complex scene and Table 7.20 presents the results of the experiments.

Figure 7.33: Complex scene 4. Left image shows the initial scene for both random grasping and grasping reflex experiments. The image on the right shows the scene after 30 grasping attempts with grasping reflex.

	random	grasping reflex
number of grasping attempts	30	30
successful grasps	0	5
unstable grasps	1	3
collisions	6	16
unsuccessful grasps	23	6

Table 7.20: The results of experiments with complex scene 4.

In the complex scene 5 (Figure 7.34, left) a lot of grasps were unstable (Table 7.21). This is because of repeated attempts of lifting object 11, which is too heavy, and object 14 where shape, weight and material all contribute to difficulty. The object narrows towards the top and is often slipped.

	random	grasping reflex
number of grasping attempts	30	30
successful grasps	0	2
unstable grasps	0	14
collisions	4	9
unsuccessful grasps	26	5

Table 7.21: The results of experiments with complex scene 5.

Figure 7.34: Complex scene 5. Left image shows the initial scene for both random grasping and grasping reflex experiments. The image on the right shows the scene after 30 grasping attempts with grasping reflex.

7.3 Discussion

The grasping reflex experiments performed on single objects as well as those performed on the complex scenes showed that there is a consistency in graspability of specific objects. In other words, some objects are grasped easily and consistently whenever they are in suitable position and image processing produces a good representation. Other objects are grasped just occasionally.

On one side this depends on how well individual object's features (weight, size, shape, colour, material) pair with type of gripper used in the experiments. On the other side it depends on how suitable the object's features are for the kind of image processing used, i.e. how difficult it is to extract good co-colour and co-planar contours. For small or distant objects, the reconstruction was often poor. In these cases images with greater resolution, or a visual attention mechanism could improve the performance.

The gripper used in current setup limits grasps of EGA 1 and EGA 2 types only to small objects. Larger objects are mostly grasped if they are concave, by the edges. Although object 9 could been grasped by the handle, it did not happen in the evaluation because the algorithm does not distinguish the handle as a good grasping place.

Table 7.22 gives the distribution of EGA grasp types for the successfully performed grasps. In single objects experiments all of the grasping types are represented. However, the same objects are grasped only with EGA 3 type of grasps in the complex scenes. In order for vertical EGA 4 type of grasp to be generated, the parent primitives would have to originate from two similar contours positioned above/bellow each other. In the five complex scenes, most objects were just partly visible and often not in optimal pose. EGA 1 and EGA

2 types of grasps have two more constrains then EGA 3 and 4 types. The parent primitives have to face each other and the their allowed mutual distance is inside a small range of values. In most situations where parent primitives originate from two separate objects, those constrains exclude grasps of type EGA 1 and 2.

	single objects	complex scenes
EGA 1	2	0
EGA 2	1	0
EGA 3	9	19
EGA 4	6	0

Table 7.22: Distribution of EGA types for successful grasps in single objects and complex scenes experiments.

The experiments showed a need for improving the criteria for ranking grasping hypotheses and the need for demanding diversity when choosing which grasps to perform in one exploration cycle. The current strategy used in the experiments led in several occasions to unsuccessful repetitive behaviour.

The current system has an open loop - "look-then-move" type of control. The drawback of this is high sensitivity to calibration errors. The accuracy of grasping operation depends directly on the accuracy of the visual sensor, the robot end-effector, and the robot-camera calibration. This could be avoided with visual servoing.

The exploration procedure could be additionally enhanced with tactile sensors and use of reactive grasping strategy. When tactile sensing and adaptive grasping behaviour are included in the system, the early grasping reflex could serve as an initial "approach" planner.

Chapter 8

Conclusion

In this master theses a system for performing an early grasping reflex was developed. It integrates image processing, grasping hypotheses generation, motion planning for collision avoidance, active collision detection using the force torque sensor and control of the robot and gripper for performing grasping exploration in a semi-unknown environment.

The experimental evaluation showed that the system is able to perform grasping even in complex environments based on a weak prior knowledge. This reflex is used as an initial behaviour in a cognitive system that aims at learning object models by exploration.

List of Figures

List of Tables

Bibliography

[Ade95] F. Ade. Grasping unknown objects, 1995.

[ASK+07] D. Aarno, J. Sommerfeld, D. Kragić, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger. Early Reactive Grasping with Second Order 3D Feature Relations. In *IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007.

[Ayd95] K. K. Aydin. Fuzzy logic, grasp preshaping for robot hands. In *ISUMA '95: Proceedings of the 3rd International Symposium on Uncertainty Modelling and Analysis*, page 520, Washington, DC, USA, 1995. IEEE Computer Society.

[BFH04] Ch. Borst, M. Fischer, and G. Hirzinger. Grasp Planning: How to Choose a Suitable Task Wrench Space. pages 319 – 325, New Orleans, LA, USA, April 2004.

[BLTK93] G.A. Bekey, H. Liu, R. Tomović, and W.J. Karplus. Knowledge-Based Control of Grasping in Robot Hands Using Heuristics from Human Motor Skills. *IEEE Trans. Robotics and Automation*, vol. 9, no. 6:709–722, 1993.

[CFMP03] Eris Chinellato, Robert B. Fisher, Antonio Morales, and Angel P. Del Pobil. Ranking planar grasp configurations for a three-finger hand. In *ICRA*, pages 1133–1138, 2003.

[CoV] Cognitive Vision Lab. http://www.mip.sdu.dk/covig/.

[CPG00] J. Coelho, J. Piater, and R. Grupen. Developing Haptic and Visual Perceptual Categories for Reaching and Grasping with a Humanoid Robot, 2000.

[Cra89] J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, 1989.

[Kal08] Sinan Kalkan. *Multi-modal Statistics of Local Image Structures and its Applications for Depth Prediction*. PhD thesis, University of Goettingen, Germany, 2008.

[KBP+] D. Kraft, E. Baseski, M. Popović, N. Krüger, N. Pugeault, D. Kragić, Sinan Kalkan, and F. Wörgötter. Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. submitted.

[Kjæ07] M. Kjærgaard. Using MicroJoysticks as Force Sensors. Master's thesis, 2007.

[KL00] J. Kuffner and S. LaValle. RRT-connect: An efficient approach to single-query path planning, April 2000.

[KLW04] N. Krüger, M. Lappe, and F. Wörgötter. Biologically Motivated Multi-modal Processing of Visual Primitives. *Interdisciplinary Journal of Artificial Intelligence the Simulation of Behavious, AISB Journal*, 1(5):417–427, 2004.

[KPK07] S. Kalkan, N. Pugeault, and N. Krüger. Perceptual Operations and Relations between 2D or 3D Visual Entities. Technical Report 2007–3, Robotics Group, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, 2007.

[Kra06] D. Kraft. Bildbasiertes Geifen mit einer Fünf-Finger-Hand. Master's thesis, In co-operation with the University of Karlsruhe, 2006.

[KW04] N. Krüger and F. Wörgötter. *Rigid Body Motion estimation for Robot Camera Calibration*, chapter Statistical and Deterministic Regularities: Utilisation of Motion and Grouping in Biological and Artificial Visual Systems, pages 131: 82–147. Advances in Imaging and Electron Physics, Elsevier Science, 2004.

[LGLM99] E. Larsen, S. Gottshalck, M. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. Technical Report TR99-018, Department of Computer Science, University of North Carolina, 1999.

[MKCA03] A. T. Miller, S. Knoop, H. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2003*, volume 2, pages 1824–1829, 2003.

[MWR] Wolfram Research Mathematica Documentation - http://documents.wolfram.com/.

[Pet] Andrew Miller Peter. GraspIt!: A Versatile Simulator for Grasp Analysis.

[PKB+08] N. Pugeault, S. Kalkan, E. Baseski, F. Wörgötter, and N. Krüger. Reconstruction uncertainty and 3d relations. *Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08)*, 2008.

[PMAJ04] Raphael Pelossof, Andrew T. Miller, Peter K. Allen, and Tony Jebara. An SVM Learning Approach to Robotic Grasping. In *ICRA*, pages 3512–3518, 2004.

[PWK06] N. Pugeault, F. Wörgötter, and N. Krüger. Multi-modal Scene reconstruction using Perceptual Grouping Constraints. In *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision*, 2006.

[RTT] The Orocos Real-Time Toolkit - http://www.orocos.org/rtt.

[RW] RobWork - http://www.mip.sdu.dk/robwork/.

[Sch] http://www.schunk.com/.

[SDK+06] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng. Learning to grasp novel objects using vision. *In 10th International Symposium of Experimental Robotics (ISER)*, 2006.

[Stä05] Stäubli Faverges. *VAL3 reference manual*, version 5.2 edition, 2005.

[TB94] Michael J. Taylor and Andrew Blake. Grasping the Apparent Contour. In *ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision*, pages 25–34, London, UK, 1994. Springer-Verlag.

[TK02] G. Taylor and L. Kleeman. Grasping unknown objects with a humanoid Robot. *Proceedings 2002 Australasian Conference on Robotics and Automation*, pages 191–196, 2002.

[WFG] David S. Wheeler, Andrew H. Fagg, and Roderic A. Grupen. Learning Prospective Pick and Place Behavior.

[WJLC05] B. Wang, L. Jiang, J.W. LI, and H.G. Cai. Grasping unknown objects based on 3D model reconstruction. *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pages 461 – 466, 2005.

Cognitive Agents – A Procedural Perspective relying on "Predictability"

F. Wörgötter[1,‡], A. Agostini[2], N. Krüger[3], N. Shylo[1] and B. Porr[4]

1) Bernstein Center for Computational Neuroscience, University of Göttingen, Göttingen, Germany, (worgott@bccn-goettingen.de)

2) Institut de Robotica i Informatica Industrial (CSIC-UPC), Barcelona, Spain (agostini@iri.upc.edu)

3) Cognitive Vision Lab, The Maersk Mc-Kinney Moller Institute University of Southern Denmark, Odense, Denmark, (norbert@mmmi.sdu.dk)

4) Dept. of Electrical and Electronic Engineering, University of Glasgow, Glasgow, Scotland, UK (porr@elec.gla.ac.uk)

Abstract: Embodied cognition suggests that complex cognitive traits can only arise when agents have a body situated in the world. The aspects of embodiment and situatedness are being discussed here from the perspective of linear systems theory. This perspective treats bodies as dynamic, temporally variable entities, which can be extended (or curtailed) at their boundaries. We show how acting agents can, for example, actively extend their body for some time by incorporating *predictably* behaving parts of the world and how this affects the transfer functions. We suggest that primates have mastered this to a large degree increasingly splitting their world into predictable and unpredictable entities. We argue that this kind of temporary body extension may have been instrumental in paving the route for the development higher cognitive complexity as it is reliably widening the cause-effect horizon about the actions of the agent. A first robot experiment is sketched to support these ideas.

In the second part of this article we discuss the concept of Object-Action Complexes (OACs) introduced by the European PACO-PLUS consortium to emphasize the notion that for a cognitive agent objects and actions are inseparably intertwined. In another robot experiment we devise a semi-supervised procedure using the OAC-concept to demonstrate how an agent can acquire knowledge about its world. Here the notion of *predicting changes* fundamentally underlies the implemented procedure and we try to show how this concept can be used to improve the robot's inner model and behaviour.

Neurons are most often sensitive to changes and not to constant stimulation. Hence, we have tried to show how the *predictability of changes* induced by an agent can be used to augment the agent's body and to acquire knowledge about the external world, possibly leading to more advanced cognitive traits.

‡ To whom correspondence should be addressed at:
Bernstein Center for Computational Neuroscience
University of Göttingen
Bunsenstr. 10
D-37073 Göttingen
Germany

Phone: 0049 551 5176528
Email: worgott@bccn-goettingen.de

Table of Contents

1 Introduction:

During the last years the European Union has invested more than 100 million Euros into the field of Cognitive Robotics and adjacent fields subsumable under bio-inspired advanced robotics. As the development of such programs rests on the ever growing scientific community in these fields, this is indicative of the fact that "a lot of people believe in it". It appears, that machines with truly intelligent, cognitive features[1] have now become within reach of research and development. This may have been largely due to the emergence of "Embodied Cognition" (EC) as a possible theoretical foundation for such R&D activities (Lakoff and Johnson, 1999; Brooks, 1999; Todes 2001; Varela, Thompson, and Rosch, 1991). Summarized in one sentence EC assumes that only machines with some kind of a body, which allows direct interactions with the world, hence, which situates these machines in their world, will be able to develop advanced (cognitive) traits (Chiel and Beer 1997; Pfeifer and Scheier 1999; Steels and Brooks 1995; Clancey 1997; Clark 1999; Todes 2001; Riegler, 2002). This notion was much influenced by Rodney Brooks, who was one of the first to explicitly state these ideas in the context of robotics work (Brooks 1986). Embodied Cognition is thus different from what has been called *good old-fashioned AI* (GOFAI), which in its extreme form supports a Cartesian attitude, treating the mind as an entity independent of and, thus, not requiring, the body (see Anderson, 2003 for a comparison between the Cartesian viewpoint and EC). This article does not intend to enter into the controversy between GOFAI and embodied cognition (first pointed out explicitly by Dreyfus, 1972, see also Brooks, 1999). For our purposes it suffices to just illuminate a little bit the germination process of EC, which has to a large degree been triggered by the notion that after all GOFAI-systems have not really become intelligent (see Brooks, 1999 for a discussion). A wealth of possible problems has been put forward for explaining this. Most influential was here the discussion of the symbol grounding problem (Searle, 1980, Harnad, 1990) and the frame problem of AI (McCarthy and Hayes, 1969; Dennett, 1984) as this had prepared the ground for the germs of EC.

[1] The term "cognition" is exceedingly ill-defined and no common agreement exist about how cognitive is cognitive. After all, also ants can build houses... We will use the term also in a wider sense but always in conjunction with human cognitive traits. Furthermore, this article is largely devoted to the question, what could be a path towards cognition. Hence we are concerned with processes and not so much with their final outcome. **Cognitive complexity is, thus, as we see it, a continuum**.

Meanwhile a large number of articles have appeared discussing, often in a controversial way, which conditions are necessary for embodied cognition. A very nice summary of this is given by Wilson (2002). She wraps up six of the most common claims on cognition found in the literature that to a large degree ask the question about the interaction between agent and world, which is a central topic also of our paper.

However, one aspect of the EC-discussion is quite puzzling in general. Most of the discussion revolves around *necessary conditions* for a cognitive agent. Little is said about what would be sufficient to drive cognitive development. Necessary conditions do not specify any on-line procedure, any ontogenetic developmental process, or any phylogenetic evolutionary mechanism that could actually drive the development of cognition. Hence, from a robotics perspective, necessary conditions are only half of the game. If you cannot show (or at least suggest) a process that leads to the germination of "something cognitive", not much has been achieved along those lines.

One possible way out of this dilemma was the idea to let robots develop similar to human infants, leading to the growing field of "*developmental robotics*" (Weng et al., 2001, Lungarella et al. 2003). For this idea, we, humans, are the proof of concept. Hence: build a robot, make it similar to a human, endow it with enough sensor-motor complexity, and with a set of useful learning algorithms and let this agent develop and learn in interaction with its world and other agents (usually its designers) and you will see the emergence of cognition. This can be done with real robots, different from the field of "*evolutionary robotics*", (Nolfi and Floreano, 2000) which attempts the same goals but must almost exclusively rely on simulations, as physical robots cannot have offspring and mutate. Both fields have their successes and increasingly complex behaviour is observed in such agents, which may some day be (or look) cognitive.

What remains frustrating about these approaches is that self-organization might indeed lead to cognition (future will show), but, we are probably none the wiser as it is exceedingly difficult and many times totally impossible to gain a deeper understanding about the final (developed or evolved) system, let alone about the dynamic processes that have led to it[2].

While developmental and/or evolutionary robotics may indeed be a way forward, we would nonetheless suggest devoting more effort to the denomination, the theoretical understanding, and the technical implementation of possible *sufficient conditions* for cognition. One key question is: Is it possible to specify some processes that may in a theoretically grounded way lead the way towards cognition in machines? To this end we would like to adopt a systems theoretical perspective on agents and their world (Ashby, 1952; McFarland, 1989; Walter, 1953), which has the advantage that its cybernetic ideology is already "very procedural" as such. In doing so some aspects of biological agents (animals and their nervous system) will be discussed, which appear to be relevant in this context.

This article is structured as follows. In general we will present several different results and ideas on the questions of embodiment, situatedness and cognition. The core

[2] Think about self-organization of neural networks as an example. Many algorithms exist for this and a wide variety of problems can now be solved by ANNs. On the other hand the theory of ANNs is mostly only developed for linear systems and it is very hard to understand more difficult ANNs in an analytic way.

thread which links them is the aspect of *Predictability* and procedures which involve predictability around which these ideas evolve. First (section 2) we would like to provide an improved systems theoretical perspective on embodiment relying on linear systems theory. By this we will define in a more rigorous way what is the body of an agent against "the world outside" (Porr and Wörgötter, 2005). This specification has direct implications on understanding the interactions between agent and world from which cognition might arise. In the next section (3) we will introduce so-called "object-action complexes" (OACs) as possible structural entities relevant for cognition[3] (Hommel et al., 2001).Then (section 4) we will suggest a process by which an agent can extend its body-image and show robot experiments for this (4.1) arguing that this might be helping the agent to develop cognitive traits (4.2). Next (section 5) we will extend the OAC concept asking which aspects of objects and actions are relevant for an agent (animal) and define "Change", "Repeatability" and "Predictability", falling back on observations from the neurosciences. We will implement these aspects in another simple robot experiment (5.1) showing how a procedure can be devised by which a machine can discover parts of its world. Finally in section 6 we will conclude this article with a discussion.

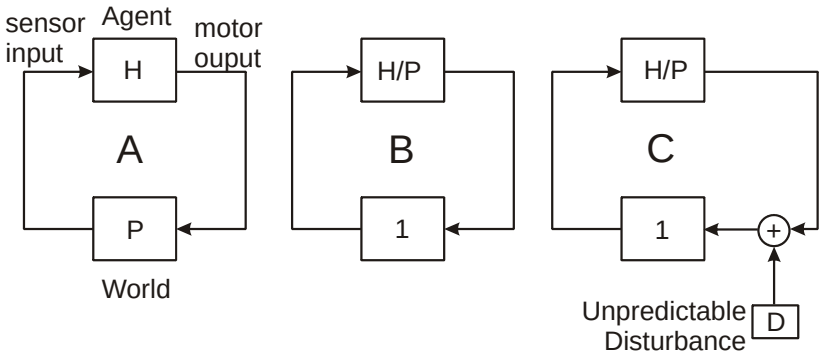


Fig. 1) Applying linear systems theory to define agent and world.

2 On Embodiment

In 2005 we had tried to provide a systems theoretical description of embodiment (Porr and Wörgötter, 2005) from the viewpoint of a constructivist (von Foerster, 1960; Maturana and Varela, 1980; von Glasersfeld, 1996). This perspective shall also be adopted here, because during phylogeny functional traits can only have developed by animals interacting with their environment. This situation is depicted by the simple diagram in Fig. 1A, where H describes the transfer function of the agent and P the transfer function of the world. Sensor inputs arriving at the agent will through H be transformed into motor outputs, while those will – in turn – be transformed into new sensor inputs for the agent through P, the transfer function of the world. For example the lifting of an object will lead to a changed visual sensation (the object moves), where this sensation is different depending on if you have lifted an object submerged in water as compared to air (different refraction index leads to different P). Note, this diagram describes in the most general sense what it means for an agent to be **situated**. The loop from agent to world and back represents in a systems theoretical diagram the notion of situatedness (Thelen and Smith, 1994; Port and van Gelder, 1995; Beer, 2000). The distinction between H and P corresponds to the distinction between "agent" (the agent's body) and "world".

[3] Note, the concept of OACs is still to some degree "emerging" and being controversially discussed (Geib et al. 2006).

At that point we had argued that every part of the world which is fully predictable could be integrated into the agent (Porr and Wörgötter, 2005). In linear systems theory this amounts to dividing the agent's original transfer function H by P arriving at a new transfer function H/P (Fig. 1B). This operation can only be performed if P is known; which is the case for all fully predictable aspects of the world. In Fig.1 B this leads to the fact that the transfer function of the world vanishes (becomes equal to one). Examples for an (incomplete) bodily integration process are the "forgetting about" a well-fitting prosthesis, which becomes much integrated in the patient's body or the feeling of a race-car pilot of "becoming united with the machine". Note, this definition (Fig. 1A) and the process of integrating P into H (Fig. 1B) do not rely on the physical world. In our sense also non-physical agents (internet robots) can be fully situated and embodied as long as they can be described by such system theoretical relations (e.g. Etzioni and Weld, 1995).

Integration of P into H, however cannot happen for unpredictable aspects of the world, which we called "disturbances" D (Porr and Wörgötter, 2003, 2005). Such disturbances can never be integrated into the body of the agent (Fig. 1C). On the contrary, unpredictable events from parts of your body can lead to the desire to remove the inflicted part (e.g. the removal of a hurting tooth). Clearly these examples are only cursory, but the mathematical distinction between predictable and unpredictable transfer functions remains valid, by which the body of an agent can in principle be set apart from the world. Note, however, **Predictability** is again only a *necessary condition* for embodiment. Clearly the trajectory of the sun is predictable but, in spite of this, the sun cannot be integrated into your body.

Two *sufficient conditions*, however, can be suggested to complete the definition.
1) Proximity: The predictable entity, for which bodily integration is to be considered, needs to be proximate (or even physically attached[4]) to the currently existing body of the agent. Most of the time this leads to the fact that the current body of the agent will be able to exert causal effects on the newly considered body parts[5]. Note, the notion of exerting a (mutually) causal influence is already captured by the division H/P, here we are asking under which circumstances such a division – the bodily integration – is allowed, for which Proximity is one sufficient condition.
2) Continuity: The new body part should be integrated for a substantial part of the life time of the agent. Hence any alteration to a body will only with time become a manifest part of the body (the body-image) of the agent. Bodies are continuous for some time.

And so we define:

> Entities which are fully predictable and proximate to the (current) body of an agent can be integrated into the body. This integration will lead to an alteration of the agent's body if it is continuous relative to the life time of the agent.

[4] The physical attachment does not have to be mechanical. Think of a WIFI connected system.
[5] Note this relation is transitive. The new body part should also causally affect the old body (if only through a load change, after having screwed on the new robot hand). In fact the new body part could be much larger than the old body. (Think of a small robot that is being physically integrated into a big plant – what is body, what is body-part?)

Hence, this definition allows understanding the body of an agent in a constructive way (much like "building" a robot). We note that it is not possible to achieve the situation of Fig.1B in full, removing the transfer function of the world entirely. Even if all entities were predictable for an agent, there would still be many that do not fulfil the two sufficient conditions, which would, thus, contribute only to P and never to H.

2.1. Critical Assessment of the Embodiment Definition

It seems that the above definition is now capturing each and everything and one might ask: are their any agents left at all that are – under this definition – non-embodied? More specifically it also seems that the definition of situatedness, by referring to closed-loop interactions between agents and world, is identical or very similar to the definition of embodiment given here.

Indeed, strong similarities do exist! Let us discuss the different possible cases:

1. NOT Situated NOT Embodied
2. Embodied AND Situated
3. Embodied NOT Situated
4. Situated NOT Embodied

Case 1 would refer to open-loop systems that are without body (e.g. pure symbol manipulation systems). This case is only of theoretical interest here as it would represent the archetype of a Cartesian attitude. Case 2 is most common for biological systems and robots and it is the case that we have discussed in Fig. 1. There is however no principle objection why this case should not hold true for more abstract A-life[6] creatures like internet agents or computer viruses as long as they obey the necessary and sufficient conditions described above within *their* world.
Case 3 can also be immediately understood as this represents all open-loop systems, which do not feed their output(s) back through the environment onto themselves. These systems are not situated. Case 4 makes troubles, it seems. How could a system be situated but not embodied? It is by the sufficient condition of Proximity and Continuity that this situation can be most easily understood and indeed, biological examples of such systems exist, which are swarms of many (embodied) individuals. In a swarm the individuals have only fleeting contact with each other. Hence as a whole the swarm represents a non-embodied (or very weakly embodied) system, which will however indeed influence its environment and also receive feedback from it. Cognitive complexity can arise from such (social) systems, for example the building of termite mounds, etc. In the context of this article we are, however, not concerned with such social, cooperative aspects, which clearly reach out into human societies, too, and will only to a minor degree discuss some implications of swarm-intelligence later.

3 Object-Action Complexes (OACs)

The question arises whether this notion of embodiment might be of any use for our understanding of agents and their cognitive development. This requires considering the processes of agent-world interaction in more detail for which we would like to introduce the concept of object-action complexes (**OAC**s). OACs had first been discussed by the European PACO+ Consortium as a possible way to better formalize the requirements for a machine to approach some level of cognitive complexity. OACs are related to state-action transitions e.g. known from machine learning (Sutton

[6] A-life=artificial life.

and Barto, 1998). They rest on the suggestion that objects and actions are inseparably intertwined. Starting with Gibson's notion of affordances (Gibson, 1979): A hollow thing with liquid may suggest drinking. For this we define an OAC formally by [O $\rightarrow^{A}$ O'], which says object O suggests action A and transforms under this action into object O' (cup-full to cup-empty) as the final outcome of this action. Note, rigorously one should define the OAC with respect to the **Attributes** (full, empty) of an object that get altered by an action. This should be kept in mind when using the abbreviated [O $\rightarrow^{A}$ O'] notation. The notion of OACs, however, goes beyond Gibson and the intertwining of Objects and Actions becomes more evident when considering the role of Actions more closely. While objects may suggest actions, it is often the action(-plan) that defines the *objectness* of a physical thing. This become clear by following example: It is the action of drinking that *makes* this thing $_{\text{hollow,full}}$ "a cup" ("a container", etc.). The decisive influence of the action becomes immediately obvious if you plan to turn the thing $_{\text{solid-bottom}}$ upside down to use it as "a pedestal" for some figurine for your mantelpiece decoration. Hence, the planned and executed action turns a thing with some (required) properties into a meaningful object. Depending on the planned actions, different properties of the same thing (hollow, full vs. solid bottom) become important. Clearly it is a very difficult (cognitive) problem for an agent to find out which properties are important and which are not. We will come to this later.

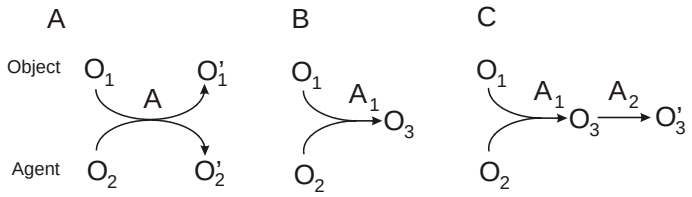


Fig. 2) Different types of transformations of objects by actions

4 Route to Cognition – Temporary Bodily Integration

In the following we would like to suggest how the above notions might be helpful in defining some processes that could indeed lead to higher behavioural complexity in an agent suggesting a possible route to (higher) cognitive traits.

It has long been known that being able to predict the world, or more specifically to predict the changes induced by the agent in the world[7], leads to improved fitness of the agent fostering its survival (and reproduction). Furthermore, a whole field has emerged during the last 10 years or so, which tried tries to explain advanced cognitive properties by so-called "probabilistic models" (Thrun, Burgard and Fox, 2005; Chater, Tenenbaum and Yuille, 2006, see also a special issue in TICS, 2006, on Probabilistic Models of Cognition). These models most often rely on Bayesian inference (Bayes, 1763; Tenenbaum, Griffiths and Kemp, 2006) which is a powerful probabilistic method for making predictions.

[7] It may make sense to point out that we are taking about Predictability from an agent centred perspective (actions by the agent). Predicting events that happen in the world without the agent's doing will also improve fitness ("thunder may predict rain"). This refers to temporally related events, which follow each other, such that this correlation can be learned. This is, however, an entirely different type of predictive mechanism not of relevance in the context of this paper.

A frame problem, however, hides here (McCarthy and Hayes 1969, Dennett 1984). In a complex world, such as that a robot or human faces, it does not make sense to try to predict each and everything. It is, thus, of interest to analyse a bit more in detail from a procedural perspective what happens when an agent interacts with the world using the OAC concept and bringing it together with some systems theory. This viewpoint will lead to the notion of "*being predictable and, therefore, we can ignore it*" as a powerful method allowing the agent to free mental resources and avoid such a possible frame problem from starters (see "Some Speculations", below).

Fig. 2A shows that during the interaction of an agent with an object normally also attributes of the agent (O_2) will change[8]. After all the effectors of an agent are also just physical objects that will be influenced when getting in touch with another object (O_1). For biological agents such contacts are most of the time fleeting and of little duration as indicated in Fig. 2A by the small contact zone of both OACs. An example would be a cat chasing a ball around. A different situation is depicted in Fig. 2B. Here a more permanent contact is established between agent and object established by action A_1. Such cases also exist for animals (a cat holds a mouse between its fangs). A new object O_3 has this way been formed, however, for most animals follow-up actions are normally very restricted and object manipulation cannot be performed (beyond the eating of the mouse). This is different for humans and in a restricted way for some animals (Hunt, 1996; Povinelli, 2000; Weir, Chappell and Kacelnik, 2002). Dexterous manipulation becomes possible by the fact that we can use the newly formed object and move it in a *predictable* way using our hands leading to situation (C) in Fig. 2. This notion is not terribly new as such but some interesting conclusions arise when looking at the situation in Fig. 2C from a systems theoretical viewpoint.

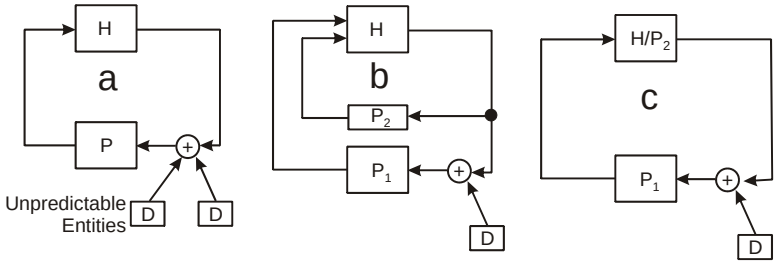


Fig. 3) Systems theoretical representation of temporary embodiment.

Fig. 3A depicts a situated agent (human) facing a few disturbances. In the process of grasping an object the human will – if successful – be able to make the grasped object fully (or at least very) predictable to her. Hence, an entity that had been a disturbance (of – say – her visual input space) will first become a predictable entity P_2 (Fig. 3B), where the human will then be able to *temporarily integrate* this entity into her body (Fig. 3C). The remaining aspects P_1 of the world cannot be integrated as they might,

[8] According to Petrick and Geib (personal communication) the aspect of an object's attribute and, thus, the OAC definition as such, needs to be more carefully considered. Think about an open door which affords the action of walking through, by which the door's attribute (open) will not change. We believe that this does not pose a problem for the definition of OAC as given above, though, as the attribute list of the object, can in principle also contain entries about the relation of the agent with respect to the object. Fig. 2A suggests that by an action the agent will also change. Or more specifically the relation of the agent to the object changes (panels B, C). As O_1 and O_2 are symmetrical, one could attach an attribute to one of them (or to both) which describes their relation to each other. Through the performed OAC, this attribute will change.

for example, be too unpredictable or too far away or from the agent's currently existing body. The idea that humans (and monkeys) indeed perform temporary bodily integration is supported by experimental results that over time cortical receptive fields are extended representing the tip of a stick, which a monkey had to use to obtain food for an prolonged period of time (Obayashi, Tanaka and Iriki, 2000). Hence a long duration, where the processes depicted in Fig.3 had taken place, has in this case even led to a long-lasting plastic change of the nervous system of this agent (monkey).

The apparently strange notion of temporary bodily integration becomes much more digestible if one thinks of an advanced robot that has grasped a pair of pliers and can handle it now with high precision and dexterity. What would prevent us – the robot's designers – from using a few screws to permanently attach these pliers to the body of the robot this way making the temporary bodily integration a permanent one?
This brings us briefly back to swarms: Here one could argue that (social) contacts formed between individuals would lead also to an augmented body concept by which a swarm can achieve more than any of its members. For slime moulds such contact can indeed be permanent and they can, indeed, form a body in the more traditional sense. Hence, it seems that gradual transitions and different types of temporary bodies do indeed exist. It would be interesting to look at swarms and swarm re-organization also from a systems theoretical perspective, but this would go beyond the scope of this article.

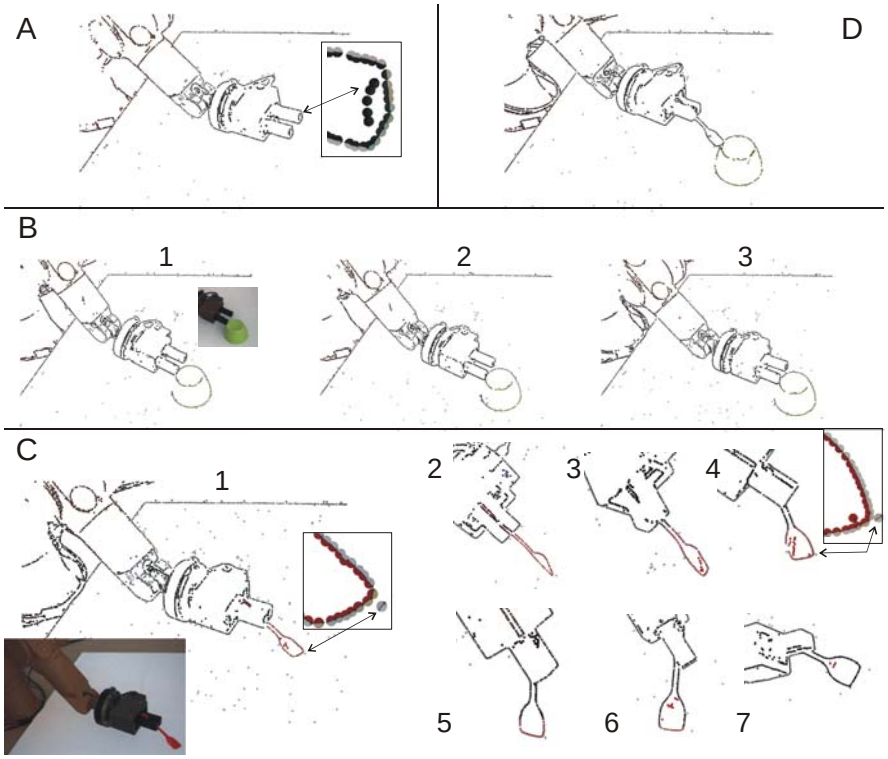


Fig. 4) Temporary embodiment experiment (for explanation see text).

4.1. Robot Experiment – Temporary Embodiment

In the following we will describe a set of experiments performed with a simple industrial robot (Stäubli, Switzerland) demonstrating how the principle of temporary bodily integration can be implemented in a machine in a simple algorithmical way to provide some support to this idea.

To this end we assume a few things for our machine to be innate:

A. A visual representation exists by which a scene can be decomposed into simple 3-D entities, which we call primitives (see Fig. 4A especially also the inset; for technical details see Krüger, Lappe and Wörgötter, 2004). Notions of distance (metric) exist therein.
B. The machine "knows" that coherently moving primitives belong together. This is known as the rigid body motion principle (see Faugeras, 1993) and corresponds to the prominent Gestalt law of Common Fate.
C. Through this, the robot has learned about its own body (gripper). This can be achieved by a purely correlation based learning process where the robot has learned to associate coherent motion in the visual field to the fact that there has been a motor command, which the machine has used to perform a movement. We assume that the process of knowing its own body is basically completed; but that this process keeps on running "in the background" to safeguard against incompleteness and errors in the body representation.
D. The machine can move its arm and it has also a certain drive to move its arm around (without which nothing would ever happen!)
E. The machine can push things around by making (visually measured) contact to entities in the scene, which do not belong to the machine. Measurement relies again on the 3-D primitives for which the concept of distance exist.
F. A grasping reflex can also be performed with some success, triggered by certain geometrical constellations between primitives from the world (Aarno et al., 2007). It can feel a successful grasp (haptic sensors) and it knows that it cannot perform another grasp without first letting go. Like babies, it, however, rather likes to hold on to a grasped entity. After some longer time it might however "get bored" and then it releases the object (also similar to small children).
G. It has an exploration drive by which it will first try to grasp a thing and if this fails (measured by the haptic sensors at the hands). It will try to push it instead. This exploration is triggered by novelty and will start as soon as something new (new primitives) are discovered in the scene.

These rather basic sub-procedural components are enough to drive the required process. Fig. 4A shows the body-representation of the robot as viewed by itself. All black-grey[9] primitives have been learned earlier (process C) to belong to its body. In the following we will for simplicity use the primitive type "black-gray" like a mental concept to graphically depict if a primitive is deemed to belong to the robot. If an object enters the visual scene the robot will try to grasp it (process G). If unsuccessful it will push it around (process E). This is shown for a not-graspable, upside down,

[9] Note, primitives at an edge are always showing two colours, one for the inside, the other for the outside. Here the robot appears dark (black) and the background brighter (grey) leading to a black-grey primitive.

green cup in Fig. 4B, where three movement stages are shown (Fig. 4B1-B3). If a grasp is successful (Fig. 4C), it will move the object (process D) like the spoon in Fig. 4C, where we show seven snapshots of movement stages. At first it will realize that the object is represented by *many* primitives which belong together (process B). This, we had at some point called "Birth of an Object" as it represents a step where the physical "object-ness" of otherwise purely visual entities (the primitives) can be ascertained (Kraft et al., 2007). If the machine does not accidentally drop the object but instead moves it for a longer time it will realize that the movement of these primitives will (albeit in a complicated geometrical way) be related to its own motor actions (process C). As it does not know better it will update its body-image based on this sensor-motor correlation and extend it to now include the coherently and predictively moving object (process C). This is shown in Fig. 4C by the gradual spread of the black-grey primitives along the spoon until the whole spoon is being re-coloured. Again we emphasize that this is just a graphical representation of the spreading inclusion of the spoon into the body image of the robot. If a new entity will enter the visual field now, sub-process G is triggered again. It feels reluctant to let go (process F) and, thus, another grasp is inhibited (also F), hence sub-processes G,E will lead to a pushing action now (Fig. 4D). As a consequence this agent, based on very primitive sub-processes, begins to perform an interaction between a very simple "tool" that extends its body (until it drops it) and the world.

Fig. 4 shows the complete experiment as performed with our robot. Clearly there are many more rather technical details that we had to take care of until the robot actually could do all this (see Aarno et al., 2007 as well as Kraft et al., 2007 for details), but the complete sequence as such does not require an other component beyond those (A-G) listed above.

4.2. Some speculations

What might have been the consequences of such a process for early humans (and possibly for nowadays robots)? Clearly, temporary bodily integration spatially extends the body of the agent and creates a totally new situatedness. As it is predictable, the agent does not have to worry about the new entity and it can largely ignore it "as such". Instead it can now use it to influence the world, thereby vastly enlarging its contact points to the world, which, before such processes came into being, had been limited to his original, non-extended body only. An agent who has realized that through such a process entities of the world can be made predictable might also have the chance to discover that it is also possible to operate with predictable outcome on other objects out in the world and that chains of predictable outcomes can be actively generated (Mendes, Hanus and Call, 2007), even possible transforming and shaping the object to a certain ends (Hunt, 1996; Weir, Chappell and Kacelnik, 2002). The discovery of the Law of Cause and Effect (Thorndike 1911) during evolution of humankind, which appears to be a cornerstone of complex cognition[10], could, thus, well have been bootstrapped by the horizon enlargement of an agent via temporary bodily integration of parts of the world.

[10] We note, that also other animals have some concept on the basic Law of Cause and Effect (many mammals, ravens, etc), but usually only of order one. Hence causal chains remain inaccessible to them.

5 Extending the OAC Concept

Above we had defined an Object-Action Complex by [O $\rightarrow^{A}$ O']. Based on this we can also define the **Change** as $\Delta O = O - O'$, where sometimes only the absolute value of ΔO is of relevance. Again we emphasize that Change is measured at the object's attributes which change. This subtractive procedure requires a memory process because we need to remember a (perceptual) Prior (O) and compare it with a Posterior (O'). Similar to the statement we had made a above, we can again point out that it is not an easy problem to determine what (which attributes) should be remembered and compared. When repeating an OAC the agent can also assess the **Expected Change,** which is $< \Delta O >$, the average Change across trials, together with its standard deviation $\sigma_{\Delta O}$ called **Repeatability**[11]. A small standard deviation would represent a high repeatability of this OAC, because all Changes are similar. Hence, the Expected Change can be considered as an **Inner Object Model** of a certain OAC. Here we need to strongly emphasize that different inner models are also possible. Change may be very relevant in general, but sometimes the absolute outcome value or a normalized one might be of greater importance for measuring the success of a task. Hence, which model to use, depends on the goals and the task of the agent.

But, which attributes are important? When performing a certain OAC many things can change: Filling a cup leads to a full, heavier cup, an emptier, lighter coffee pot a splashing noise and possibly a change of the illumination (because someone else has switched a light on). Normally, through repetition the agent can find out which properties change causally (Thorndike 1911), hence in way correlated to the OAC (certainly not the illumination), and this way the agent can improve the Expected Change reaching smaller values of $\sigma_{\Delta O}$ with more and more trials[12]. Here we note that we have tacitly assumed that the agent will be able to perform "fairly optimal" actions. Hence, the filling-action as such ought not to introduce additional contingencies which affect Change and Expected Outcome. In reality – say for small children – this in not necessarily the case. Hence such "clumsy" agents need to improve their actions in parallel to updating the Expected Outcome of the corresponding OAC. Thus, **Action Models** *as well as* Inner Object Models need to be updated and improved in parallel. Assuring convergence of such a double-procedure is far from trivial and we will discuss some aspects later.

Storing Expected Change and Deviation of course requires also some kind of memory. A simple way to measure the **Unpredictability** P of an OAC is to calculate P= $abs(\Delta O - < \Delta O >)$, Change for a single trial minus Expected Change[13]. If your actions are ok, and your inner object model is complete you should find that actual action outcome and expected outcome match. Hence, this OAC is then highly predictable and P is small. Humans keep on trying when an attempted OAC keeps on leading to an unpredictable outcome. On the other hand, they are getting bored and stop repeating actions for which a predictable outcome is again and again obtained.

[11] Note: As everyone is used to associate a small standard deviation to "good" and a large one to "bad", we will keep on calling $\sigma_{\Delta O}$ the Repeatability and not the Unrepeatability (but compare to "Unpredictability" defined later.)

[12] Alternatively a teacher can tell the agent to "pay attention to" a certain change (similar what we often do with our kids), which is a much more efficient procedure as compared to many required repetitions in trial and error learning.

[13] Note, we define (Un-)Predictability as predicting change, not predicting status-quo.

Boredom[14], hence comes with high Repeatability which corresponds to a small value of $\sigma_{\Delta O}$. One more thing should be reasonably assumed. Agents should not average Change and Predictability over their complete life span, Forgetting – hence limiting the averaging window – will help to remove the influence of early trials on the inner model (which are most of the time much more erroneous than later trials).

The notions above heavily rely on the aspect of Change, Expected Change and Predictability. The nervous system of (probably all) animals is highly change sensitive and vastly neglects constant inputs as almost all neurons respond in an adaptive way ("phasic") bringing their activity levels back to (near) zero if their inputs remain constant. Hence, without over-stating, it seems fair to say that Change is the most relevant aspect of the world for biological agents.

5.1. Robot Experiments – Discovery by Doing

The considerations above suggest that Predictability will allow an agent to extend its body, but it will also allow for "discovering the world" in a reliable way. This has been done in another set of experiments with a much simpler robot that has learned some rules and inner models of cause-effect relations by interacting with a human teacher. Here we cannot describe the full algorithmic procedure (see Agostini et al., 2008 for details), but the underlying idea is that the robot simultaneously learns cause-effects relations and rules while experiencing situations in accordance to a goal oriented behaviour. The cause-effects experienced are dictated by the rules and supervised by a teacher. The learned cause-effect pairs represent a particular coding of the model of the world strongly related to the concepts of OACs. In this work, we would like to show how the procedure of learning cause-effects can be general delineated in terms of the concepts of Change, Predictability, etc, developed above. Thus, we need to define a rather general process for the discovery and accumulation of knowledge by an agent. Clearly this process does not have to be the only one, rather it is meant as an example that the consequent employment of the above discussed principles could lead to a smarter agent. In the following we will not discuss how an agent would make a choice (*decision making*), this is a different issue unrelated to the model building for OACs. The same is true for *planning*, which also is not considered here. Hence, in a sense we are strictly dealing with a procedure for interactive OAC learning and model update.

Table 1: Definitions

ΔO	Change
$< \Delta O >$	Average Change
$\sigma_{\Delta O}$	Repeatability
P	Unpredictability
Φ_S	Success Threshold
Φ_B	Boredom Threshold
N,B	Counters

Let us come back to our definition of OAC, Change, Expected Outcome, and especially Unpredictability $P = abs(\Delta O - < \Delta O >)$, Change for a single trial minus Expected Change[15]. As the definition of P relies only on (cumulative) changes it is independent of the specific OAC performed and we can measure the **Success** of any OAC by holding P against some threshold Φ. If $P < \Phi_S$ the OAC was successful.

[14] It is equally possible to use these quantities to define other psychologically relevant entities, like Frustration, which would arise if Repeatability does not get better.

[15] We note that all these definitions apply to one given OAC. Any other OAC will have values for these entities on its own. This would require an index, which we would like to avoid for easier writing and more clarity.

Note: All variables, counters, etc. are without index, but are always specific for the one OAC to which they belong!
Note: N counts how often a given OAC has been performed during the life-time of the experiment; B counts how often it has been performed *one after the other* (in a sequence).

1: Initialize everything
2. OAC: Perceive situation
3: Match against Memory and select all possible performable OACs and keep them for possible use below (code lines 11+)

The following lines (4-10) assess the result from the last-performed OAC
4: If $\sigma_{\Delta O} < \Phi_B$ or B large (**Boredom**) (Agent was getting bored by the last trial)
5: if existing, select any performable OAC different from last
6: else Goto NOAC (there are no OACs known to be performable, ask for a new one!)
7: Else
8: If N small and P> Φ_S (we had a **Failure** last trial but are not sure if this was chance or not)
9: if performable again, select the same OAC (try again to rule out chance)
10: else continue (same OAC is in this situation not performable, try another existing or new)

In all other cases: Exploration
11: Else
12: If some performable OACs exist
13: Choose OAC freely (normally in a task oriented way)
14: Else {
15: NOAC: Ask for a new OAC (a new OAC is given by the supervisor)
16: Set N= B= ΔO = $<\Delta O>$ = 0 and $\Phi_S = \Phi_B = \sigma_{\Delta O}$ = large (Initialize) }

17: If current OAC different from previous OAC
18: Set B=0 (Repetition counter which assesses Boredom)
19: Perform OAC
20: Increase counters B and N
21: Measure ΔO (**Change**)
22: Calculate P=abs(ΔO - $< \Delta O >$) (**Unpredictability**)
23: If P< Φ_S (**Success**) {
24: Update $< \Delta O >$ and $\sigma_{\Delta O}$ **(Model Update)**
25: Lower Φ_S (Be more demanding on Success next time!)
26: Change Φ_B (if desired)
27: Goto OAC }
28: Else (**Surprise**) (don't do any model update, something is "funny")
29: Goto OAC
30: End

Note: The selection of OACs can (but does not have to) be performed so as to try to resolve a previously experienced surprise, which is the case for the PFR in Fig. 5 which resolves a surprise that prevented a previously attempted "Go" OAC. The procedure is supervised to the degree that the agent will indicate if none of its OACs can be performed (code line 3) in a given situation (code line 2). The agent will in this case ask for an instruction.

Cleary, success thresholds should change with praxis. An inexperienced young child building a LEGO toy house will have a different (in this definition: higher) success threshold than an experienced adult[16]. Especially, when performing an OAC for the

[16] These definitions are strictly following a constructivist attitude and, hence, rely entirely on the agents own interactions with the world, where we assume that minimization of Repeatability $\sigma_{\Delta O}$ is the agent's target function for improving on an OAC. Clearly other target functions could be imposed on the agent also by ways of a teacher (hence "from the outside"), which would however, not alter the arguments

first time, its success threshold should be very high, because the agent does not know about any Expected Change. **Failure** occurs for $P > \Phi_S$ and this may lead to **Surprise** and the triggering of a different OAC, possibly for resolving the Surprise, or it may lead to the repetition of the current OAC to check the consistency of the Failure. Thus, Surprise arising from an unpredictable situation can be one strong driving force for the discovery of the world by a machine. Note, Surprise is not the only driving force. Even without Failures (which lead to Surprise) exploration can be triggered by Boredom**.** If Repeatability is too good, hence $\sigma_{\Delta O} < \Phi_B$, or if an OAC has been performed too often one after the other the agent should be getting bored, which would trigger another OAC. All this suggests now a procedure for an agent for acquiring knowledge about the world as shown in the box. Together with the bodily extension discussed above the agent could then especially also acquire knowledge about the world beyond the reach of its own effectors. In the process of performing this iteration between Model Update and Exploration, the agent will acquire improved models for the OACs it has tried; hence, it will learn about some cause-effect relations. If bored, it will try out something new and it will, possibly by the help of a teacher, try to resolve Surprises. Clearly this procedure can be augmented by many facets and such a "discover-by-doing" process is not limited to the above described pseudo-code.

Several things need to be discussed. For example, it is necessary to comment on the statement "….to resolve the Surprise…" brought up above. Many times surprise will arise if an agent faces a presumably known situation (a cup) but with an as yet undiscovered attribute (closed). A well-tried OAC (cup-filling) will now fail, triggering surprise. The agent could in this case indeed (via trial and error or by being taught) try to discover that only Cup_{open} can be filled. This way it would resolve the surprise. One also needs to be aware that we are assuming that the agent has a certain reliable action repertoire. This assumption would however not really hold for babies where the action repertoire is limited and unreliable. How can they learn? After all, such inept agents face a two-fold difficulty: They have to update their inner object model and they need to improve their actions at the same time. This, however, is an ill-posed problem. At least one of the two (inner object model or action) ought to be fairly reliable, without which the agent cannot update the other as it would not know from where a contingency arises. How can human babies, who are faced with this dilemma, solve this problem? We believe that their bodily limitations are the answer. Their limited action repertoire and their very limited reach leads in the beginning to only very simple cause-effect relations with very wide allowance for the resulting effects. As a consequence of these imposed limitations they will stumble across OACs "regardless of what they are doing". Hence, the constraining element comes from their bodies and from the specific very restricted situatedness in which they find themselves. We would argue that these constraints are probably enough the make a simultaneous action improvement and inner object model update convergent.

In the context of this article the complete "Discovery-by-Doing" procedure cannot be described, but we can show results for a robot that is supposed to learn the rules of how to clear a path in front of it by pushing boxes away. Note, as this is meant to exemplify the concepts, the logic in the following example is not fully rigorous and complete (see Agostini et al., 2008 for a fully operational version of this). Initially, the

---

presented in the following. We note that self-estimation of statistical correlation properties, like through $< \Delta O >$ and $\sigma_{\Delta O}$, is a very time-consuming process (trial and error learning). Other forms of learning, which provide a target function from the outside (supervised learning) are quicker, but the target function may not match the agents own goals.

robot assumes that it can move forward regardless of any visual input (OAC: "Go" forward, code lines roughly 2,3,11,13,17-22). It will update its inner model of this OAC (in vision space, code lines 23→24-27), which, however, will be just an empty space as moving over an empty space will in this case lead also into an empty space.

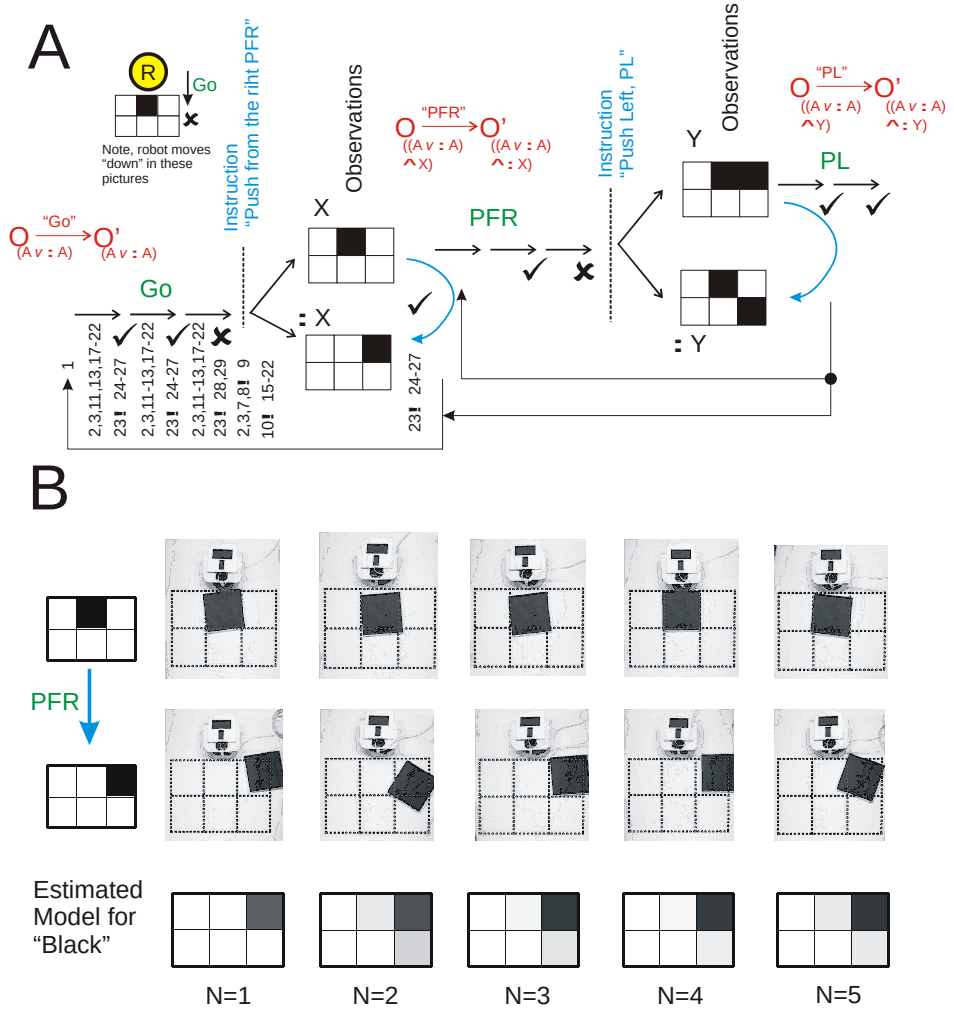


Fig. 5) Different OACs for a box-pushing robot. A) Schematic, B) Five trials of the PFR action. ***A)*** *Abbreviations: Red denotes an OAC where the subscript represents the attribute list with: ¬ logical NOT, v OR, ^ AND. The attribute (Av¬A) stands for "anything". Above the red arrow stands the action of the OAC (adopting the robots view point: "Go", go forward; "PFR", push box from the right sideways; "PL", push left box downward). The symbol √ stands for success, while ✗ denotes a failure. Small numbers below individual trials refer to the code lines in the pseudo code (see box). The dashed vertical lines denote Surprise as a consequence of failure. X, Y are arbitrarily denoted observable attributes that change their state (blue arrow) following actions PFR or PL. The long black back-arrows denote that the robot can, after having cleared one obstacle, return to any of the previous OACs, in accordance to the acting criterion (policy, instructed, etc). Furthermore, for the "PL" situation it is not relevant if attribute X or ¬X exists, hence X does not show up in the "PL"-OAC.* ***B)*** *Shown are perceptual priors and their posterior after the PRF action. The gradual development of the inner model is depicted at the bottom, for graphical reasons, however, here given by absolute grey values and not by Change.*

However, it will eventually encounter a Surprise whenever something blocks its way (the go-forward OAC falls below its success threshold, code lines 23→28,29). As the surprise is genuine (code lines 2,3,7,8→10), the human instructs the robot to perform a certain (pre-programmed, hence reflex-like) action called "push the obstacle from

the right" by which the obstacle can be removed (code lines 15). The robot will perform the new action, measure the outcome (code lines 16-22), record it into a new inner model storage area (23→24-27), and update the policy coding by generating rules in accordance to this new OAC. The robot is then artificially forced by us to move back to its staring position. If it does not see anything it will perform a "go-forward" OAC, if it encounters the same blocker again it will now weakly assume (hence with high threshold Φ_S) that the same situation allows for the same "push-from-right" OAC which will lead to success and to the removal of the blocker.

Hence, if it tries this for the second time the machine will average the new outcome with the previously received result and this way it will arrive at a better inner object model (Expected Change) at the same time threshold Φ_S will be lowered. In panel B of Fig. 5 we show five real experiments of the robot performing the PFR action. Cleary neither the initial situations are identical nor are the outcomes. The development of an inner model for the outcome of PFR is shown at the bottom. To provide an easier intuition we are using here absolute grey value averages O and calculate <O> instead of changes ΔO to calculate < ΔO>. Note, as emphasized above different types of inner models may be calculated by an agent (absolute, relative, normalized, etc., etc.) depending on the task.

6 Conclusion

Many of the notions put forward in this article had been presented at least in parts by other authors, which we have tried to acknowledge along the way. So this paper hopes to contribute to the discussions on embodied cognition by its different (systems theoretical) perspective as well as by the attempt to find an uninterrupted procedure, based on the evaluation of Change and Predictability, towards more cognitive complexity. "Uninterrupted" means here that the same principles have probably been valid for our primordial ancestors and are still applicable for us and our children (which we teach). It is plainly impossible to draw all cross-links to prior work and some aspects (like Piaget's view, which had just also surfaced a bit, Piaget, 1930) have been totally left out.

Thus, in this article we have tried to provide a procedural perspective on embodiment and cognition using ideas from linear systems theory to explain our assumptions. This Ansatz allows disentangling the concept of embodiment from that of situatedness and relies heavily on concepts of "Change" and "Predictability", which are prevalent in neuronal responses. Specifically, we tried to show what happens to a system when an agent is able to manipulate an object in a predictable way: From a systems theoretical viewpoint this object then becomes temporarily integrated into the agent's body. By ways of simple robot experiments we have shown that the idea of temporary bodily integration can be consistently represented on a machine using the principle of rigid body motion (RBM) to integrate entities into the body image of the agent as soon as those entities move coherently and if this happens together with a motor command that the agent has produced. Given preconditions A-G above, this process relies then only on signal correlations and does not need any teacher or other external influence (which could not have been there anyways during evolution).

In the following we had discussed how a process (relying on Change, Predictability, Surprise and Boredom) can be formulated by which a robot can (with help) discover the rules of its world. This is achieved by exploration, repetition, surprise and the resolving of surprise, which can either be achieved by a teacher (like in Fig. 5A

"Instruction") or through trial and error learning. Trial and error learning makes such procedural trees accessible already early in human evolution, where supervision has still not played a big role and learning was almost exclusively by trial and error.

7 Acknowledgements

This work was funded by the European project PACO-PLUS. Thanks are due to Mila Popovic for her help with the robot experiment in Figure 4 and to T. Asfour, C. Geib, B. Hommel, R. Petrick, M. Steedman, S. Wischmann, and the rest of the PACO-PLUS consortium for many stimulating and fruitful discussions.

8 References

Aarno, D., Sommerfeld, J. Kragic, D., Pugeault, N. Kalkan, S., Wörgötter, F. Kraft, D. and Krüger, N. (2007). Early reactive grasping with second order 3D Feature. IEEE International Conference on Robotics and Automation (ICRA), Workshop; From features to actions – Unifying perspectives in computational and robot vision.

Agostini, A., Celaya, E., Torras, C. and Wörgötter, F. (2008) Action rule induction from cause-effects pairs learned through robot-teacher interaction.

Anderson, M.L. (2003). Embodied Cognition: A field guide. Artificial intelligence, 149: 91-130.

Ashby, W. R. (1952). Design for a Brain, Chapman and Hall, London.

Bayes, T. (1763): "An Essay towards solving a Problem in the Doctrine of Chances". See http://www.stat.ucla.edu/history/essay.pdf

Beer, R.D. (2000). Dynamical approaches to cognitive science. Trends in Cognitive Sciences, 4, 91-99.

Brooks, R. (1986). A robust layered control system for a mobile robot. Journal of Robotics & Automation, 2, 14-23.

Brooks, R. (1999). Cambrian Intelligence: The early History of the New AI, MIT Press, Cambridge, MA.

Chater, N. Tenenbaum, J.B. and Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. Trends in Cognitive Sciences, 10, 7:287-291.

Chiel, H. and Beer, R. (1997). The brain has a body: Adaptive behaviour emerges from interactions of nervous system, bdy, and environment. Trends in Neurosciences, 20, 553-557.

Clancey, W.J. (1997). Situated Cognition: On Human Knowledge and Computer Representations, Cambridge University Press, Cambridge.

Clark, A. (1999). Embodied, situated and distributed cognition, in: Bechtel, W. and Graham, G. (Eds.), A Companion to Cognitive Science, Basil Blackwell, 506-517

Dennett, D.C. (1984): Cognitive wheels: The frame problem of AI, in: Hookaway, C. (Ed.), Minds, Machines and Evolution, Cambridge University Press, Cambridge, 129-151.

Dreyfus, H.L. (1972). What Computers can`t do: A Critique of Artificial Intelligence, Harper and Row, New York.

Etzioni, O. and Weld, D.S. (1995) Intelligent agents on the internet: Fact, fiction, and forecast. IEEE Expert, 4(10), 44-49.

Faugeras, O.D. (1993) Three-dimensional Computer Vision. MIT Press.

Geib, C., Mourao, K., Petrick, R., Pugeault, N., Steedman, M., Krüger, N. and Wörgötter, F. (2006). Object Action Complexes as an Interface for Planning and Robot Control. IEEE RAS Int Conf. Humanoid Robots(Genova):Dec. 4-6, 2006.

Gibson, J.J. (1979). The Ecological Approach to Visual Perception, Houghton Mifflin, Boston. (Currently published by Lawrence Eribaum, Hillsdale, NJ.)

Harnad, S. (1990) The Symbol Grounding Problem. Physica D 42: 335-346

Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001). The theory of event coding (TEC): A framework for perception and action planning. Behavioral and Brain Sciences, 24, 849-878.

Hunt, G.R. (1996) Manufacture and use of hook-tools by New Caledonian crows. Nature, 379: 249-251.

Kraft, D., Bacseski, E., Popovic, M., Krüger, N., Pugeault, N., Kragic, D., Kalkan, S. and Wörgötter, F. (2007) Birth of the Object: Detection of Objectness and Extraction of object shape through object action complexes. (in press)

Krüger, N., Lappe, M. and Wörgötter, F. (2004). Biologically Motivated Multi-modal Processing of visual primitives. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour. 1,5:417-428.

Lakoff, G. and Johnson, M. (1999). Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought, Basic Books, New York.

Lungarella, M., Metta, G., Pfeifer, R and Sandini, G. (2003). Developmental robotics: a survey. Connection Science, 15,4:151-190.

Maturana, H. and Varela, F.J. (1980). Autopoiesis and cognition: the realization of the living. Reidel, Dordrecht.

McCarthy, J. and Hayes, P.J. (1969) Some philosophical problems from the standpoint of artificial intelligence. Mach. Intell., 7, 195-204.

McFarland, D. J. (1989). Problems of Animal Behaviour. Harlow: Longman Scientific & Technical.

Mendes, N. Hanus, D. and Call, J. (2007). Raising the level: Orang-utans use water as a tool. Biology letters. Doi:10.1098/rsbl.2007.0198 (published online)

Nolfi, S. and Floreano, D. (2000). Evolutionary Robotics, The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press/Bradford Books)

Obayashi, S., Tanaka, M. and Iriki, A. (2000). Subjective image of invisible hand coded by monkey intraparietal neurons. NeuroReport 11, 3499-3505.

Pfeifer, R. and Scheier, C. (1999). Understanding intelligence. Cambridge, MA. MIT Press.

Piaget, J. (1930). The child's conception of physical causality. Routledge and Kegan Paul (Publ.)

Porr, B. and Wörgötter, F. (2003). Isotropic Sequence order learning. Neural computation, 15:831-864.

Porr, B. and Wörgötter, F.(2005). Inside embodiment – What means Embodiment for Radical Constructivists? Kybernetes 34:105-117

Port, R.F. and van Gelder, T. (1995). Mind as motion: Explorations in the dynamics of cognition. Cambridge, MA. MIT Press.

Povinelli, D.J. (2000). Folk physics for apes. Oxford Univ. Press, Oxford.

Riegler, A. (2002). When is a cognitive system embodied? Cogn. Syst. Res., 3:339-348.

Searle, J.R. (1980). Minds, brains and programs. Behavioral and Brain Sciences, 3: 417-457.

Special Issue: Trends in Cognitive Sciences (2006), 10,7.

Steels, L. and Brooks, R. (1995). The artificial life route to artificial intelligence: Building embodied, situated agents. Hillsdale, NJ: Erlbaum.

Sutton, R.S. and Barto, A.G. (1998). Reinforcement Learning: An Introduction. Bradford Books, MIT Press, Cambridge, MA, 2002 edition.

Tenenbaum, J.B., Griffiths, T.L. and Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. Trends in Cognitive Sciences, 10, 7:309-317.

Thelen, E. and Smith, L.B. (1994). A dynamic systems approach to the development of cognition and action. Cambridge, MA. MIT Press.

Thorndike, E.L. (1911) Animal intelligence. New York: Macmillan.

Thrun, S., Burgard, W. and Fox, D. (2005). Probalistic Robotics, MIT Press.

Todes, S. (2001). Body and World, MIT Press, Cambridge, MA.

Varela, F., Thompson, E. and Rosch, E. (1991). The Embodied Mind, MIT Press, Cambridge, MA.

von Foerster, H. (1960). On self-organizing systems and their environments. In Yovits, M. and Cameron, S., editors, Self-Organizing Systems, Pergaman Press, London, 31-50.

von Glasersfeld, E. (1996). Learning and adaptation in constructivism. In Smith, L., editor, Critical Readings on Piaget, Routledge, London and New York, 22-27.

Walter, W.G. (1953). The Living Brain. G. Duckworth, London.

Weir, A.S., Chappell, J. and Kacelnik, A. (2002). Shaping of Hooks in New Caledonian Crows. Science, 297: 981.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M. and Thelen, E. (2001). Artificial Intelligence: Autonomous Mental Development by Robots and Animals. Science, 291, 5504: 599-600.

Wilson, M. (2002). Six views of embodied cognition. Psychonomic Bulletin & Review. 9, 4:625-636.