Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D27
Deliverable Title :	Multi-platform grasp controller and grounding of grasping primitives
Type:	PU
Authors	V. Kyrki, J. Laaksonen, J. Ilonen, E. Nikandrova, A. Morales, J. Felip, J. Bernabe, T. Asfour, M. Przybylski, J. Schill
Contributing Partners	LUT, UJI, KIT

Contractual Date of Delivery to the EC:	28-02-2012
Actual Date of Delivery to the EC:	28-02-2012

Contents

Chapter 1

Executive summary

Deliverable D27 presents fourth year developments within workpackage WP3 "Self-experience of Grasping and Multimodal Grounding". According to the Technical Annex of the project, D27 presents activities connected to Tasks 3.1, 3.2, and 3.3. The objectives of these tasks are defined as

- [**Task 3.1**] **- Control Architecture.** Initially, a hierarchical control architecture will be defined and developed such that it allows relating the concepts of the grasping ontology defined in WP2 to the immediate control. After the architecture has been defined, this task will continue with the definition and development of the general control architecture components, mainly a Cartesian controller and high-level supervisory and visual controllers.
- [**Task 3.2**] **- Multimodal Grounding.** The task aims for the definition and development of a grounding mechanism connecting action primitives and attributes with uncertain sensor information, including modelling of the uncertainties involved. Initially, the modelling of uncertainties of the three sensor types (visual, tactile, proprioceptive) is studied considering the context of the attributes of the grasping ontology. Later, the task will continue by studying the temporal grounding problem as a state estimation problem with uncertain information, as the concepts and therefore the symbol set are defined by the grasping ontology.
- [**Task 3.3**] **- Robust action primitives.** The task aims for the definition and evaluation of adaptive and robust control approaches for individual action primitives. The main focus will be on studying the possible grasp primitives for different hand kinematics (parallel jaw, three-fingered, five fingered) and to identify robust parameterisable primitives through evaluation. Parameterisation of the primitives allows self-experience to be used for improving the performance during future attempts.

The work in this deliverable relates to the following fourth year milestone:

- [**Milestone 11**] - Integration and evaluation of scenarios on multiple experimental platforms, demonstration of cognitive capabilities of robots.

The progress in WP3 is presented briefly below, and in more detail in the appendix containing attached scientific publications and reports.

- Attachment A presents work describing the primitive based manipulation paradigm developed in the project, related to Task 3.1. Moreover, the work proposes a complete set of reactive sensor-based manipulation primitives for object transport, related to Task 3.3. The work extends and combines results shown in Year 2 and Year 3 deliverables D13 and D20. We demonstrate the completion of an object transportation task on two different platforms using the same abstract description. We also demonstrate that a complex task of emptying a box filled with many objects can be solved using the paradigm.
- Attachment B presents work in reconstructing object 3-D shape by fusing information from visual and tactile sensors, related to Task 3.2. To reconstruct an object, a low-quality point cloud model

is first created based on stereo from a single view point. This initial model is used to plan a grasp on the object, which is then executed with a gripper equipped with tactile sensors. The object model is then refined based on the contacts detected by the tactile sensors, and further actions can be decided based on the refined model.

- Attachment C presents a study of how much object information can be extracted from tactile exploration, related to Task 3.2. Using a simulator as an internal model (memory) of the robot, the evaluation is based on assessing how much information error minimization between predicted and actual sensor readings can provide about the environment. The focus in the study is an object transportation task and experiments indicate that a single exploration action is not guaranteed to provide much information for all uncertain factors if the attempt is not originally planned with information gain in mind.

- Attachment D presents work extending the results of grasp stability recognition presented in Year 3 deliverable D20 to sensor-based grasp planning under uncertainty, related to Tasks 3.2 and 3.3. The work presents a novel probabilistic framework for grasping, in which grasp and object attributes, online sensor information and the stability of a grasp are all considered through probabilistic models. The framework is demonstrated by building the necessary probabilistic models using Gaussian Process regression, and using the models with an MCMC approach to estimate a target object's pose and grasp stability during grasp attempts. The framework is also demonstrated on a real robotic platform.

- Attachment E presents work describing an approach for estimating contact between robot fingers and an object using only visual input, related to Task 3.2. The approach is based on the assumption that object motions are caused by the contacts. The approach is validated through experiments which show that the visual contact estimation is able to detect contacts in a scenario where the contacts could not be perceived using tactile sensors. In addition, the ability to detect contacts when the hand is visually occluded is demonstrated.

- Attachment F presents an extension of the grasp stability recognition, which estimates the stability continuously during a grasp attempt, related to Task 3.2. The approach is based on temporal filtering of a support vector machine classifier output. Experimental evaluation is performed on an anthropomorphic ARMAR-IIIb. The results demonstrate that the continuous estimation provides equal performance to the earlier approaches while reducing the time to reach a stable grasp significantly. Moreover, the results demonstrate for the first time that the learning based stability estimation can be used with a flexible, pneumatically actuated hand, in contrast to the rigid hands used in earlier works.

Appendix A

Attached papers

A Javier Felip, Janne Laaksonen, Ville Kyrki, and Antonio Morales. Manipulation Primitives: A Paradigm for Abstraction and Execution of Grasping and Manipulation Tasks. Submitted to *IEEE Robotics and Automation Magazine, Special issue on Mobile manipulation.*

B Jarmo Ilonen and Ville Kyrki. 3-D Object Reconstruction by Fusion of Visual and Tactile Sensing. Submitted to *Robotics: Science and Systems, RSS 2012.*

C Ekaterina Nikandrova and Ville Kyrki. What do contacts tell about an object? Submitted to *IEEE International Conference on Biomedical Robotics and Biomechatronics, BioRob 2012.*

D Janne Laaksonen and Ville Kyrki. Probabilistic Sensor-based Grasping. Draft, to be submitted to *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012.*

E Jose A. Bernabe and Antonio Morales. Contact detection and location from robot and object tracking on RGB-D images Submitted to *Robotics: Science and Systems, RSS 2012.*

F Julian Schill, Janne Laaksonen, Markus Przybylski, Ville Kyrkim Tamim Asfour, and Rüdiger Dillmann. Learning Continuous Grasp Stability for a Humanoid Robot Hand Based on Tactile Sensing. Submitted to *IEEE International Conference on Biomedical Robotics and Biomechatronics, BioRob 2012.*

Manipulation Primitives: A Paradigm for Abstraction and Execution of Grasping and Manipulation Tasks

Javier Felip, Janne Laaksonen, Ville Kyrki, Antonio Morales

***Abstract*—Robot grasping and manipulation in unstructured service scenarios is a challenging scientific and engineering problem to a large extent caused by incomplete and uncertain knowledge about the environment. Sensor-based reactive and hybrid approaches have proven a promising line of study to address these issues. However the use of sensor-based approaches is difficult in situations where knowledge transfer between embodiments is desired, because the approaches are usually tightly coupled to a particular embodiment.**

This paper proposes a paradigm for modelling and execution of manipulation actions, which makes knowledge transfer between embodiments possible, while retaining the capabilities of the individual embodiments. The paradigm is built upon the concept of manipulation primitives, which constitute a vocabulary of atomic actions. More complex actions, such as object transport, can then be described as sequences of abstract primitives, the set of which is shared over different embodiments. These abstract models can then be translated to embodiment specific models, constituting of reactive sensor-based controllers, such that the full capabilities of each platform can be utilised.

The paradigm is demonstrated by presenting a practical implementation, including a description of the translation mechanism and a complete set of primitives for object transport. Moreover, we demonstrate the completion of an object transportation task on two different platforms using the same abstract description. Finally, we also demonstrate that a complex task of emptying a box filled with many objects can be solved using the paradigm.

I. Introduction

Robots operating in unstructured service scenarios, for example mobile manipulators, need to operate robustly despite incomplete and uncertain information about their environment. Seminal works on reactive control [1], [2] demonstrated that the use of several low-level perception/actuation loops enabled robots to adapt to unknown scenarios. These approaches were soon extended by incorporating high-level planners prioritizing the available reactive behaviours, giving birth to hybrid deliberative/reactive control [3], [4].

In complete contrast to reactive approaches, manipulation and grasping has been traditionally addressed through planning of contact states. Methods, such as grasp quality metrics based on form and force closure, are very powerful when the uncertainty in robot and environment models is minimal. However, in service robotics manipulation scenarios, uncertainties appear in many quantities, for example, inaccurate knowledge about the poses of objects and obstactles, incomplete models of object shape and physical properties, or inaccurate kinematics in flexible robots. To address these issues, there are a few works using the reactive paradigm. One of the earliest works in reactive grasping proposed the use of a light beam sensor to align the gripper with an unknown object [5]. More recently, solutions such as IR proximity sensors [6], tactile sensors [7], [8], and force and tactile feedback [9] have been proposed. In contrast to traditional grasp planning, these approaches aim to adapt the robot hand to the shape of the target object reactively instead of placing contacts in planned locations. As a consequence, exact models of objects are not required, which is a great strength of the approaches.

Reactive manipulation approaches are usually, however, specific to a particular embodiment, which makes it difficult to transfer plans between different embodiments and even from humans to robots. This paper presents a paradigm of manipulation primitives, which combines the idea of reactive control with action abstraction. The paradigm describes manipulation tasks in terms of atomic primitives, which offers several advantages. Firstly, complex actions can be described in terms of simple abstract primitives. Secondly, plans can be shared over different embodiments because the vocabulary of primitives is shared. Finally, these abstract models can be translated to embodiment specific models, constituting of reactive sensor-based controllers, such that the full capabilities of each platform can be utilised.

Next, we continue by presenting related work in Sec. II and the terminology used in Sec. III. The paradigm is demonstrated by presenting an implementation, including a description of abstraction and translation in Sec. IV and a complete set of primitives for object transport in Sec. V. Moreover, we demonstrate the completion of an object transportation task on two different platforms using the same abstract description in Sec. VI-A. Finally, we also demonstrate solving a complex task of emptying a box filled with many objects using the paradigm in Sec. VI-B.

II. Related work

Few studies have addressed the issue of abstracting hardware from action. Petersson and Christensen presented a somewhat similar framework in [10] but to our knowledge that framework has never been demonstrated in practice with multiple embodiments. Earlier version of the work presented here appeared in [11]. Finally, Ellenberg et al. studied how algorithms for humanoid robot walking can be transferred between embodiments [12]. To our knowledge, the work presented here is the most advanced in the context of abstracting action across multiple embodiments.

J. Felip and A. Morales are with Robotic Intelligence Laboratory, Department of Computer Science and Engineering, Universitat Jaume I, 12006 Castellón, Spain {jfelip,morales}@uji.es.

J. Laaksonen and V. Kyrki are with Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland {jalaakso,kyrki}@lut.fi.

The idea of control primitives is not new in robotics, and particularly in robot grasping. Earlier works propose individual control primitives for different problems such as to control a hand [13], to define object movements [14] and its relations [15] and to control a manipulator [16]. Despite different definitions of primitives, all of them present a common trend, discretizing and reducing the complexity of controlling a robotic setup by reducing the search space for planning. Other similar approaches include Object Action Complexes [17] and the physical interaction framework of [18]. However, in contrast to this work, all of the above consider primitives which are specific to a particular embodiment.

An alternative approach to address the problem of unknown environment is to use sensors, for example vision, to build the necessary models. Vision has been used to obtain the shape of unknown target objects [19], [20] and to determine the location and pose of objects [21]. In both cases, visual input was used to plan feasible grasps. Visual feedback can also be used during reaching for an object. Murphy et al. [22] uses visual techniques to correct the orientation of a four-finger hand while approaching an object to improve contact locations. Once contact between object and robot has been reached, tactile and force sensors can be applied. Tactile measurements can be used to estimate the quality of grasps [23], [24], [25], [26] or the shape of an object [27] with the purpose of reaching better contact locations through a sequence of grasping/regrasping actions. Contact information can also be used to program complex dexterous manipulation operations like finger repositioning while holding the object [23], [28]. Several works have combined the use of several sensors to complete the process of grasp planning and execution [29], [30].

III. Manipulation primitives

We define a *manipulation primitive* as a single reactive controller designed to perform a specific primitive action on a particular embodiment. Each primitive is parameterized to allow it to be used in different situations. A focused control policy which uses the available sensor feedback is then used to achieve predefined success or failure conditions.

Primitives are the elementary symbols of a vocabulary that is used to describe *actions* and *tasks*. A *task* is a semantically meaningful goal, such as emptying a grocery bag, consisting of one or more *actions*. Each action describes a single manipulation action, for example, moving an object from one location to another, as a Finite State Machine (FSM) where the states correspond to manipulation primitives. The transitions between states are triggered by *events*, predefined perceptual or internal conditions.

IV. Abstract tasks and primitives

Primitives are by definition embodiment specific. However, embodiments with similar capabilities allows the definition of primitives with similar behaviour and purpose, which can be thought as *abstract manipulation primitives*. The focused purpose of primitives simplifies the development of equivalent primitives on several embodiments. This equivalence also enables the transmission and execution of plans between different embodiments. The abstract manipulation primitives can then be used to describe *abstract actions*. We call this abstract representation of an action the Abstract State Machine (ASM).

A. Abstract State Machine

The abstract state machine is a hardware independent description of a manipulation action. The ASM uses XML (eXtensible Markup Language) to describe the relevant information, such as the states and transitions of the state machine. Also information about the target object, e.g. its pose and mass, and obstacles in the manipulation environment are described. All properties and definitions in XML are hardware independent.

The abstract state machine is described through definition of states and transitions between the states. Both states and transitions have properties that can be used to further inform of the intended action. For example, the hand preshape for grasping or the path for the end-effector can be set through state properties. The transition properties describe the conditions when the transition is triggered. For example, the loss of a grasp can trigger a transition to another state.

In addition to the properties, the state also has type attributes, which describe the manipulator motion that is desired from each defined state. These attributes are:

- **move:** Moving the manipulator without an object.
- **transport:** Moving the manipulator with an object.
- **grasp:** Grasping an object.
- **place:** Placing an object.
- **release:** Releasing an object.
- **slide:** Sliding an object.
- **success:** Indicating end state with success.
- **failure:** Indicating end state with failure.

These attributes are the key factor in selecting the primitive controllers during the translation process depicted in IV-B. An example abstract state machine and its XML definition, describing a simple grasp and lift manipulation, is shown in Fig. 1. Some of the elements have been left out for brevity, e.g. properties of the object and some of the common transitions, e.g. timeout to the failure state.

B. Translation from ASM to FSM

The translation process is what combines the abstract state machine and the embodiment specific state machine (FSM). The translation takes the abstract state machine as an input, and translates the abstract state machine into an embodiment specific state machine. The translation process is depicted in Fig. 2.

As can be seen from Fig. 2, the translation component needs input defining the configuration of the translation process, i.e., the target platform and the platform specific transitions and primitive controllers used directly in the embodiment specific state machine. The benefit of this arrangement is that the only hardware dependent blocks shown in the figure are the primitive controllers and transitions that are platform specific.

```
<statemachine>
  <state name="approach" type="move">
    <movement>free</movement>
    <hand_shape>open</hand_shape>
  </state>
  <state name="preshape_hand" type="move">
    <movement>guarded</movement>
    <hand_shape>pinch_grasp_preshape</hand_shape>
  </state>
  <state name="grasp_object" type="grasp">
    <movement>guarded</movement>
    <hand_shape>pinch_grasp</hand_shape>
  </state>
  <state name="lift_object" type="transport">
    <movement>guarded</movement>
    <hand_shape>pinch_grasp</hand_shape>
    <path>
      <position>0.2 0.6 0.25</position>
    </path>
  </state>
  <state name="success_end" type="success">
  </state>
  <state name="fail_end" type="failure">
  </state>

  <transition origin="approach"
    destination="preshape_hand">
    <success/>
  </transition>
  <transition origin="preshape_hand"
    destination="grasp_object">
    <success/>
  </transition>
  <transition origin="grasp_object"
    destination="lift_object">
    <success/>
    <grasp_stable/>
  </transition>
  <transition origin="lift_object"
    destination="fail_end">
  <grasp_lost/>
  </transition>
  <transition origin="lift_object"
    destination="success_end">
    <success/>
    <grasp_stable/>
  </transition>
</statemachine>
```

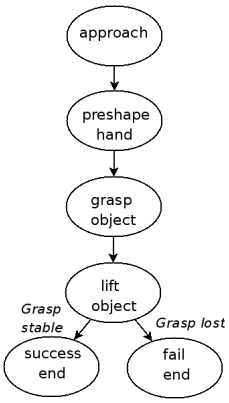


Fig. 1. An abstract state machine.

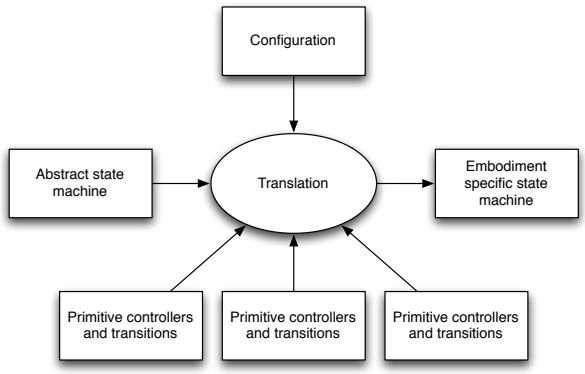


Fig. 2. Translation process.

Also the critical requirement of real-time operation for sensor-based control is fulfilled as the embodiment specific state machine can be run as is, without any additional overhead from maintaining hardware independence.

The translation process also requires a mapping component which produces the embodiment specific state machine from the abstract automaton. The mapping itself is developed manually, but once the mapping component is complete, the translation process from any ASM is performed automatically. This mapping is fairly simple to implement as there are only a limited amount of input properties and the mapping is aware of only of each individual primitive and transition of an abstract action.

Furthermore, a common Cartesian velocity control interface is defined for the arm, thus, we can use primitive controllers that use the arm velocity control for all hardware platforms without modifications. The same applies to some transition conditions, e.g. timeout can be used in all platforms. Thus, building the basic primitive controllers and transitions gives the added benefit of not having to implement all controllers and transitions for each new platform introduced to the system.

V. Manipulation Primitives for Object Transport

In this section we describe a set of sensor-based manipulation primitives for object transport, corresponding to the abstract types described in the previous section. The primitives are described independent of a particular hardware platform but a set of control and sensor requirements shown in Table I are needed to implement them. All the primitives are parametrizable, requiring one common parameter: an approach vector to the object. All other parameters are optional and shown in the table.

A. Grasp primitive

A grasp primitive can be implemented just by closing the hand. It has, however, been demonstrated that using sensor based methods the success rate of this primitive can be increased significantly [9]. We propose to implement the primitive using a sensor based controller that performs several corrective movements in order to get a stable grasp. These movements are divided into three phases: alignment, sliding grasp and force adaptation.

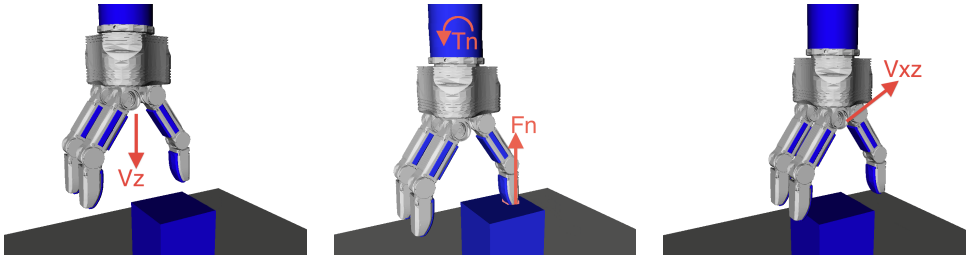
The optional parameters for the implemented grasp primitive are the pregrasp type (cylindrical, spherical, hook) and size. The corrective movements to be performed can also be configured using a binary parameter that tells the controller whether to use a correction phase or not. By default all the correction phases are applied.

1) Alignment: In some situations, the initial approach vector is not pointing to the center of the object, and thus there is a premature collision during the approach phase. This contact can be detected using a force-torque sensor mounted on the wrist. Using the torque, the contact point is estimated and a correction is performed to center the object. An example of this is depicted in Fig. 3. The contact can also detected using tactile sensors triggering the centring behavior. Alignment correction improves grasping of objects with location uncertainty by allowing the hand to align its center with the object.

2) Sliding grasp: When approaching, the hand usually makes contact with the supporting surface instead of the object (See Fig. 4(a)). In this case, closing the hand can result in unsuccessful grasps especially for small objects. To counter this problem, sliding grasp correction is used. The corrective movement consists of moving the hand forwards or backwards depending on the force sensed along its Z axis while the fingers are closing (see Fig. 4) to maintain stable, light contact with the supporting plane. When the fingers are no longer able to close, because the object is grasped or the fingers reach their joint limits, the sliding grasp control ends. The correction

Primitive name	*Other parameters*	*Control requeriments*	*Sensor requeriments*
Grasp	Pregrasp size, grasp preshape	Arm cartesian velocity controller	Wrist force-torque sensor, tactile sensors
Lift	Lift height	Arm cartesian position control	None
Transport	Obstacles, trajectory, constraints	Arm cartesian velocity control	None
Place	Contact treshold	Arm cartesian velocity control	Wrist force-torque sensor
Release	Hand position	None	None

TABLE I
VOCABULARY OF PRIMITIVES, PARAMETERS AND REQUERIMENTS.



(a) Arm moving towards the object. (b) Contact generates torque in the wrist. (c) Correction movement is performed.

Fig. 3. Grasp primitive: Alignment phase.

improves grasping small objects by sliding the fingers on the supporting plane until the object is securely grasped.

The behavior of this correction phase is shown in Fig. 4. The hand starts closing and when the fingers make contact with the surface, the force they are applying is detected in the wrist, thus the arm moves back (Fig. 4(a)). The fingers continue closing and because no contact force is detected, the arm moves forward (Fig. 4(b)). In Fig. 4(c) the fingers are not able to close anymore and the primitive ends successfully.

3) Force adaptation: The force of the fingers is increased to improve grasp stability. The primitive ends with a success if at the end the object is still in the hand, detected with joint angles or contact information.

B. Transport primitive

The purpose of the transport primitive is to move the arm to a specified target position while the hand holds an object. The primitive can also be used to move the arm without an object.

The trajectory to move the arm from the starting point to the target can be constrained by specifying optional parameters. A trajectory can be specified as a list of joint positions that define the state of each joint during all the transport primitive execution. A less restrictive constraint is to specify the end-effector Cartesian trajectory. Instead of defining the exact trajectory that the robot must follow, it is also possible to specify position, velocity or acceleration limits. A force-torque is used to stop the movement if a collision is detected.

Optional parameters can also be used to describe environment obstacles as an obstacle point cloud, in which case a force-field [31] based collision avoidance strategy is used to generate a collision free trajectory from current to target position maintaining the hand orientation.

For instance, if the task is to transport a mug full of water without pouring the liquid, acceleration should be constrained

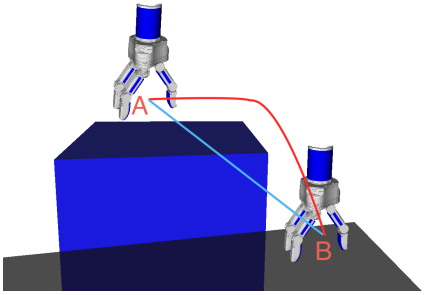


Fig. 5. Example of execution of the constrained transport primitive from the starting point (a) to the target point (b). Red line: Standard trajectory. Blue line: Position constrained trajectory.

to a low value on all axes and the rotation velocity of the table plane axes should be set to 0 to prevent tilting the mug. If the target position cannot be reached without breaking the specified constraints, the primitive ends with a failure. In Fig. V-B an example of a position constrained trajectory is shown. The convex hull of the box is defined as forbidden space to define position constraints.

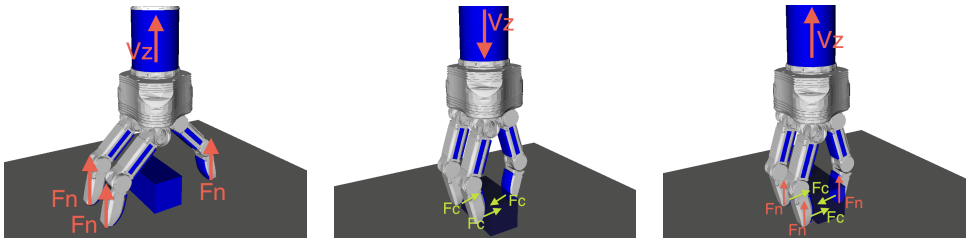
C. Place primitive

The place primitive is used to place an object on a supporting plane while detecting the support on-line using sensor feedback. The arm moves down until a contact is detected with a force sensor. When a force opposing the movement direction is felt it assumes that the object is placed.

This primitive can be configured with an optional parameter defining the force threshold needed to detect a contact. An example execution of this primitive is shown in Fig. 6.

D. Release primitive

Releasing an object can be difficult because the fingers can, while opening, collide with the supporting plane or other parts

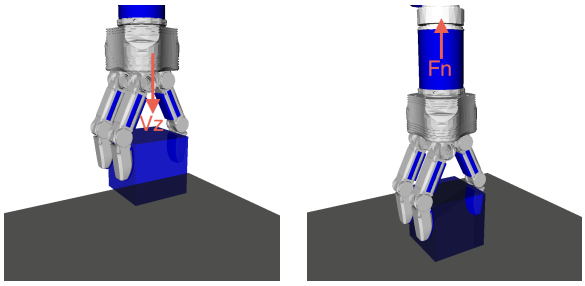


(a) The fingers contact the table while closing. Thus the controller sets the velocity to move the hand back.

(b) The fingers are closing and the contact with the table is lost. Vz is set forwards.

(c) The hand contacts the table again but the object is already grasped.

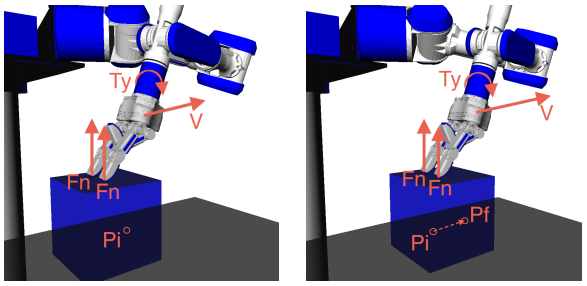
Fig. 4. Grasp primitive: Sliding grasp phase.



(a) Arm moving the object towards the surface.

(b) Contact is detected by force/torque sensor.

Fig. 6. Place primitive.



(a) From the starting position with a hook preshape, the arm moves down until it touches the object, then it starts moving towards the target.

(b) The object slides over the table from Pi to Pf. The primitive keeps the applied force stable.

Fig. 8. Slide primitive.

of the object (see Fig. 7(a)). To handle this problem, the release primitive opens the hand slowly while the arm moves back. The movement of the arm is force-controlled and the arm only moves back if there is a contact detected between the opening fingers and the surface. The sequence of movements is shown in Fig. 7. This primitive can be configured by setting the target hand position after release.

E. Slide primitive

The purpose of the slide primitive is to push an object from the top and slide it on a surface, as shown in Fig. 8. Using force control the arm applies a desired force (Fn) to the object, then moves towards a set target, keeping the applied force constant (Fig. 8(a)). The contact fixes the arm and object movement allowing the robot to slide the object on the surface from the starting to the target position (Fig. 8(b)). Only the target position is a required parameter, but the applied force can be configured by setting a desired force range.

VI. Demonstrations

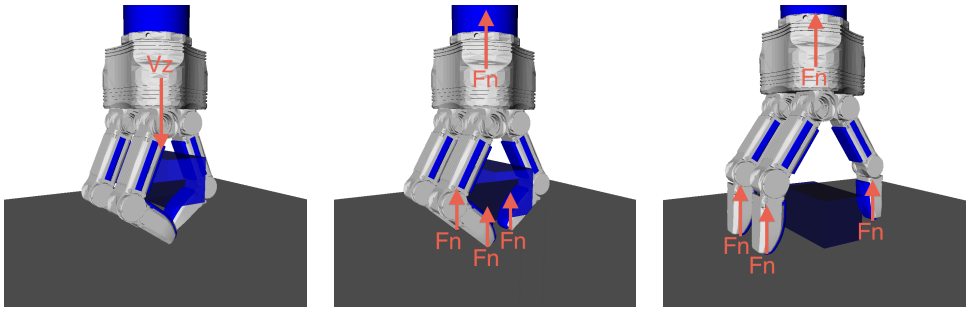
To demonstrate the applicability of the manipulation primitives paradigm, we present two demonstrations: mapping of actions for different embodiments and completion of a complex task using the paradigm and the primitives described. Our main experimental platform is *Tombatossals*, an anthropomorphic torso with 23 DOF shown in Fig. 10. The platform is composed of two 7 DOF Mitsubishi PA10 arms. The left arm has a 4 DOF Barrett Hand and the right arm a 7DOF Schunk SDH2 Hand. Both hands are endowed with Weiss Robotics tactile sensors on the fingertips. Each arm also has a JR3 force-torque sensor mounted on the wrist. Visual system of the platform is composed of a TO40 4 DOF pan-tilt-verge head with two Imaging Source DFK 31BF03-Z2 cameras and a Microsoft Kinect.

A. Action mapping to different embodiments

We demonstrate the mapping of the abstract state machine by developing a simple pick and place abstract state machine. To enable mapping of the ASM, we implemented the translation component described in Section IV-B for two different platforms, *Tombatossals* and a Melfa RV-3SB 6-DOF arm with a PG70 parallel jaw gripper equipped with Weiss tactile sensors. The implementation included the required platform specific controllers for the different states in the ASM and the platform specific transitions, as well as the required configuration information.

As a result, shown in Fig. 9, we were able to grasp objects based only on the sensor data from the hand and the arm, when given estimate of the pose of the object. Using the same abstract state machine for both platforms shows clearly that we are able to use abstraction and then turn this abstract information to platform specific primitives and transitions used in the sensor-based control.

In the context of the demonstration we used the same Cartesian controllers for both arms. On the other hand, the hands are too different in terms of kinematics and sensors so that each hand had its own implementation of control.



(a) Hand before opening the fingers.

(b) The hand cannot release the object, the fingers are blocked by the surface. The normal force in each finger propagates to the wrist.

(c) The hand moves back and continues opening the fingers. The object is released successfully.

Fig. 7. Release primitive.

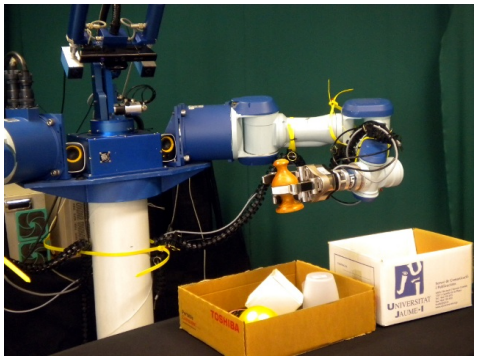


Fig. 10. The experimental robotic platform: Tombatossals, the UJI humanoid torso.

Also the transitions for grasp stability or instability were customized for each of the platforms in order to effectively use the different sensor capabilities available on the platforms. It should be noted that the task was nevertheless described using only the abstract description, without any embodiment specific information.

B. *Emptying a box: Execution of a complex task*

To demonstrate that the paradigm is valid for executing complex sensor-based tasks, we chose the task of emptying a box with no previous information about the number, location and pose of the objects inside. More precisely, the assumptions are that the object positions inside the box are not restricted, objects can be in any position and orientation inside the box, except that the is some clearance between the objects and the sides of the box. The object size is defined by the SDH2 hand dimensions so that the objects fit inside the hand and are thus graspable. The box is set on an even plane inside the arm workspace. Tombatossals is used as the experimental platform.

The task is solved using a *pick and place* loop executed for each object. This loop consists of a sequence of primitives structured and executed as a Finite State Machine (FSM) as described earlier. The FSM included several manipulation primitives which are instantiated to direct the robot to pick up an object from a starting position and place it to a destination position. The FSM is shown in Fig. 11. The required parameters are the starting approach vector to a target object and the target position to place it. This procedure is repeated until the box is empty.

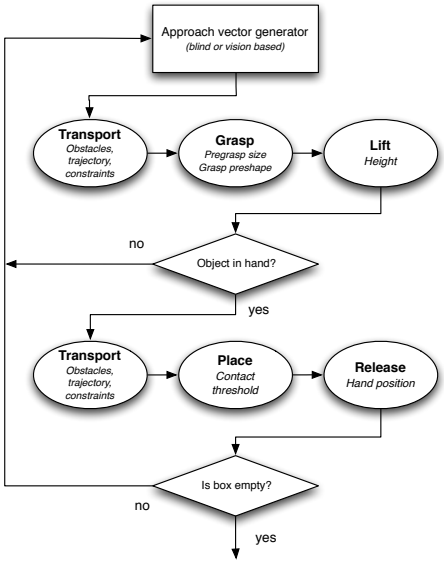


Fig. 11. State machine for a pick and place task. Primitives are represented by circles. External processes are depicted using boxes. Diamond boxes represent conditions that are checked inside the parent primitive to determine the next transition. Inside each primitive, some examples of parameters are written in italics.

A key part of this loop is the generation of initial approach vectors. Three strategies were implemented: *random blind*, *blind exploration* and a *vision-based* method. In the first one, top-grasp approach vectors are generated uniformly at random inside the known location of the box. In this case, ending the whole process is decided by a human observer.

In the *blind exploration* strategy, the arm moves down until a contact is detected. If the contact is an object the approach vector is generated over that contact location. If the contact is the box bottom the hand starts moving along the bottom

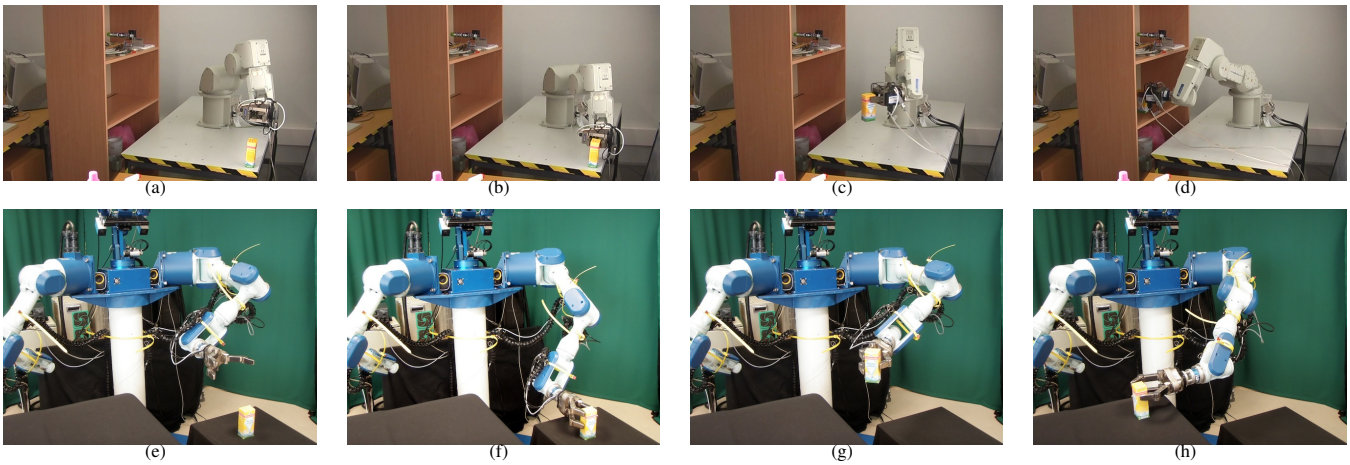


Fig. 9. Action execution on different platforms, (a)-(d) Melfa RV-3SB with PG70, (e)-(h) *Tombatossals*: (a,e) Approach; (b,f) Grasp; (c,g) Transport; (d,h) Release.

until it detects a contact using the tactile and the force-torque sensor. As the position of the box is known, propioception is used to determine whether the contact is with an object or with the box bottom. The exploration trajectory followed by the hand is shown in Fig. 13(a). The task ends after exploring the whole box without finding an object.

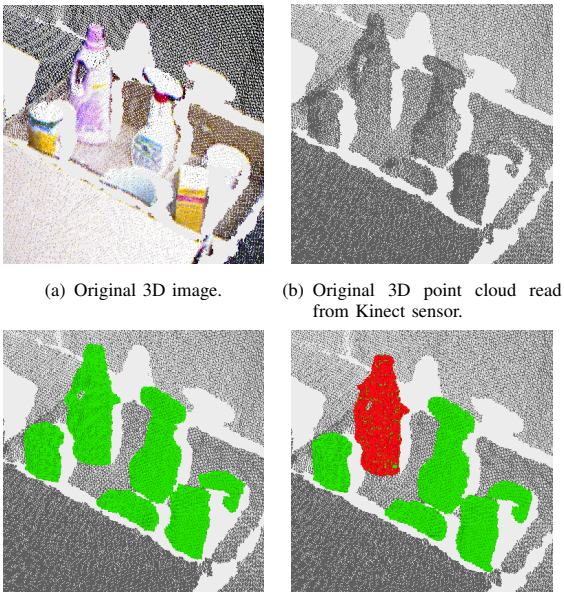
In the *vision-based* strategy, the Kinect sensor is used. This sensor outputs a depth image and an RGB image. Objects are segmented from the environment using a pass-trough filter using the box boundaries and clustered as shown in Fig. 12. The approach vector is determined to approach the centroid of a randomly chosen cluster. The task ends when no clusters are left.

In order to validate the approach we carried out a total of 30 experiments of emptying a box filled with five unknown objects (see Fig. 13(b)). 10 experiments were performed for each approach vector generation method. All the methods were able to empty the box successfully 10 times out of 10. However, several attempts were sometimes needed to grasp an object. The number of attempts needed to lift an object was recorded. Fig. 14 shows the average number of required attempts depending on the number of objects remaining in the box as well as the standard deviation.

It is evident from the figure that the vision-based approach vector generation improves the results over the blind methods, which is hardly surprising. However, the interesting result is that the blind methods were also able to complete the task successfully every time.

VII. Discussion and Conclusion

Over the years, robot grasping has split to two different approaches. On one hand, object-based robot grasping which focus on considering a grasp as a set of locations on the object shape, through which manipulation forces are exerted on the object. On the other, hand-based approaches rely on the capabilities and constraints of the robot embodiment, focusing on control aspects. The manipulation primitives paradigm



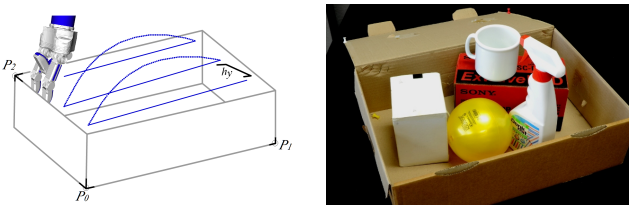
(a) Original 3D image.

(b) Original 3D point cloud read from Kinect sensor.

(c) Virtual box background filtering. Background points are colored in gray and objects are in green.

(d) Object clustering and selection. Background points are marked in gray, objects in green, and the selected cluster is labeled in red

Fig. 12. 3D point cloud segmentation phases.



(a) Hand preshape for exploration and exploration trajectory.

(b) A possible object layout

Fig. 13. Exploration trajectory and object layout.

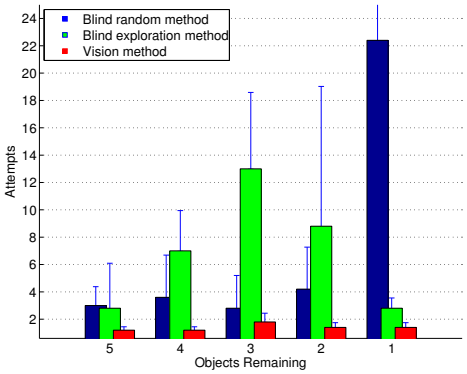


Fig. 14. Average and standard deviation of required attempts depending on the number of objects remaining. The standard deviation for the blind random method when there is only one object left is truncated in the picture, its value is 39.32

belongs to the latter approach, considering grasps as starting conditions for the action and letting the control loop and the real world itself guide the execution. The demonstrations shown indicate that manipulation problems can be solved in complex, unstructured scenarios while retaining hardware independence on a higher level.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement ICT-215821.

REFERENCES

[1] R. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14 – 23, mar 1986.

[2] J. Connell, "A behavior-based arm controller," *Robotics and Automation, IEEE Transactions on*, vol. 5, pp. 784 –791, dec 1989.

[3] J. Connell, "Sss: a hybrid architecture applied to robot navigation," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2719 –2724 vol.3, may 1992.

[4] R. C. Arkin, *Behavior-Based robotics*. The MIT Press, 1998.

[5] M. Teichmann and B. Mishra, "Reactive algorithms for grasping using a modified parallel jaw gripper," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1931 –1936 vol.3, May 1994.

[6] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *IEEE International Conference on Robotics and Automation*, pp. 2098 –2105, May 2009.

[7] K. Hsiao, S. Chitta, M. Ciocarlie, and E. Jones, "Contact-reactive grasping of objects with partial shape information," in *IEEE International Conference on Robotics and Automation*, 2010.

[8] D. Gunji, Y. Mizoguch, S. Teshigawara, A. Ming, A. Namiki, M. Ishikawa, and M. Shimojo, "Grasping force control of multi-fingered robot hand based on slip detection sing tactile sensor," in *SICE Annual Conference, 2008*, pp. 894 –899, August 2008.

[9] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,*, pp. 1811 –1816, October 2009.

[10] L. Petersson, M. Egerstedtt, and H. Christensen, "A hybrid control architecture for mobile manipulation," in *Proc. IEEE/RSJ IROS'99*, pp. 1285–1291, 1999.

[11] J. Laaksonen, J. Felip, A. Morales, and V. Kyrki, "Embodiment independent manipulation through action abstraction," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Anchorage, USA), 2010.

[12] R. Ellenberg, R. Sherbert, P. Oh, A. Alspach, R. Gross, and J. Oh, "A common interface for humanoid simulation and hardware," in *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pp. 587 –592, dec. 2010.

[13] T. Speeter, "Primitive based control of the utah/mit dextrous hand," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 866 –877 vol.1, Apr. 1991.

[14] P. Michelman and P. Allen, "Forming complex dextrous manipulations from task primitives," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3383 –3388 vol.4, May 1994.

[15] J. Morrow and P. Khosla, "Manipulation task primitives for composing robot skills," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 3354 –3359 vol.4, Apr. 1997.

[16] Y. Hasegawa, M. Higashiura, and T. Fukuda, "Object manipulation coordinating multiple primitive motions," in *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, vol. 2, pp. 741 – 746 vol.2, July 2003.

[17] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krger, and F. Wrgtter, "Object action complexes as an interface for planning and robot control.," in *HUMANOIDS-06 Workshop Toward Cognitive Humanoid Robots*, 2006.

[18] M. Prats, P. Sanz, and A. del Pobil, "A framework for compliant physical interaction," *Autonomous Robots*, vol. 28, pp. 89–111, 2010. 10.1007/s10514-009-9145-8.

[19] A. Morales, P. Sanz, A. del Pobil, and A. Fagg, "Vision-based three-finger grasp synthesis constrained by hand geometry," *Robotics and Autonomus Systems*, vol. 54, pp. 496–512, June 2006.

[20] 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 *IEEE Conference on Robotics and Automation (submitted*, 2007.

[21] P. Azad, T. Asfour, and R. Dillmann, "Combining appearance-based and model-based methods for real-time object recognition and 6D-localization," in *International Conference on Intelligent Robots and Systems*, (Beijing, China), 2006.

[22] T. Murphy, D. Lyons, and A. Hendriks, "Stable grasping with a multi-fingered robot hand: A behavior-based approach," in *IEEE/RSJ International Conference on Robotics and Intelligent Systems*, vol. 2, (Yokohama, Japan), pp. 867–874, July 1993.

[23] J. Coelho Jr. and R. Grupen, "A Control Basis for Learning Multifingered Grasps," *Journal of Robotic Systems*, vol. 14, pp. 545–557, October 1997.

[24] R. Platt, A. H. Fagg, and R. Gruppen, "Nullspace composition of control laws for grasping," in *IEEE International Conference on Robots and Intelligent Systems*, (Lausanne, Switzerland), pp. 1717–1723, 2002.

[25] T. Mouri, H. Kawasaki, and S. Ito, "Unknown object grasping strategy imitating human grasping reflex for anthropomorphic robot hand," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 1, no. 1, pp. 1–11, 2007.

[26] Y. Bekiroglu, J. Laaksonen, J. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.

[27] P. Allen and K. Roberts, "Haptic object recognition using a multi-fingered dextrous hand," *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 342–347 vol.1, May 1989.

[28] M. Huber and R. Grupen, "Robust finger gaits from closed-loop controllers," *IEEE/RSJ International Conference on Robotics and Intelligent Systems*, vol. 2, pp. 1578–1584 vol.2, 2002.

[29] P. Allen, A. T. Miller, P. Oh, and B. Leibowitz, "Using tactile and visual sensing with a robotic hand," in *IEEE International Conference on Robotics and Automation*, (Albuquerque, New Mexico), pp. 677–681, Apr. 1997.

[30] B. J. Grzyb, E. Chinellato, A. Morales, , and A. P. del Pobil, "Robust grasping of 3D objects with stereo vision and tactile feedback," in *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, (Coimbra, Portugal), pp. 851 – 858, 2008.

[31] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 500 – 505, Mar. 1985.

3-D Object Reconstruction by Fusion of Visual and Tactile Sensing

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

***Abstract*—Manipulation and grasping of unknown objects is one of the great challenges for general purpose service robotics. There are vision based methods for grasping unknown objects, but further planning of manipulation using these approaches is difficult, because the methods do not generally create a 3-D model of the object. On the other hand, 3-D reconstruction of an object from a single view is only possible with additional assumptions.**

In this work, we propose to reconstruct a 3-D model of an object by fusion of visual and tactile information. Assuming the object is symmetric, a low-quality point cloud model is first created based on stereo. This initial model is used to plan a grasp on the object, which is then executed with a gripper equipped with tactile sensors. The object model, including the symmetry parameters, can then be refined based on the contacts detected by the tactile sensors, and further actions can be decided based on the refined model. The main contribution of this work is an optimal estimation approach for the fusion of the visual and tactile data applying the constraint of object symmetry. The fusion is formulated as a state estimation problem and solved with iterative extended Kalman filter. The approach is validated experimentally using both artificial and real data.

I. INTRODUCTION

Manipulation and grasping of unknown objects is one of the great challenges for general purpose service robotics. Vision based methods for grasping unknown objects exist (e.g. [16, 14, 10, 12]), but generally such methods do not create a 3-D model of the object, which nakes their use in further planning of manipulation difficult. In most real-world scenarios, grasping is only a prerequisite action for further object manipulation and a 3-D object model is needed for manipulation planning especially when the manipulation happens in a constrained space and obstacle avoidance is important.

3-D reconstruction using a single view or a narrow baseline system is only possible with additional assumptions. In this paper, we consider the case of objects with symmetry, which is not a severe limitation in service robotic scenarios, as many of the everyday objects are in fact symmetric. In this respect, the work is based on the idea presented in [5] where the symmetry assumption was used to create a grasp plan for unknown objects.

Besides vision, tactile exploration and probing can be used to generate 3-D models of objects. However, this approach is limited to exploring only a small local area at a time, and thus the generation of a full 3-D model is often too slow for practical applications.

In this paper, we present a method for reconstructing a symmetric 3-D object by fusion of visual and tactile information. Because of the complementary nature of the two senses a more complete model can be created by fusing the senses compared to what is possible using visual or tactile information alone. Moreover, the model is created while the unknown object is grasped and no additional exploratory probing actions are necessary.

The main contribution of this work is an optimal estimation approach for the fusion of the visual and tactile data applying the constraint of object symmetry. The fusion is formulated as a state estimation problem and solved with iterative extended Kalman filter. The model is first initialized from vision and a symmetry constraint is applied to create a model for the complete object. From this initial model a grasp plan is devised and the object is grasped with a robotic hand with tactile sensors. The object model is then refined based on the contacts detected by the tactile sensors. By considering the estimation in a statistical framework we are able to optimally combine the information from the two sensors. The approach is demonstrated and validated experimentally using both generated and real data.

II. RELATED WORK

Visual reconstruction has been studied widely, but limiting the reconstruction to a narrow baseline stereo view, a common limitation in service robotics, has the problem that only the visible frontside of the object can be reconstructed, which does not generally enable object grasping or other manipulation. Some constraints must be applied to extrapolate from a single frontal view the non-visible 3-D structure. Some examples are reconstruction of building indoor models by applying the constraint of orthogonal planes [11], reconstruction of surface of revolution objects [6], of symmetric piecewise planar objects [18], or of objects with a symmetry plane [5]. The approach proposed in this paper extends the idea of [5] by using tactile measurements to refine both 3-D points as well as the symmetry parameters.

On the other hand, tactile exploration can reveal the whole structure of an object, but only for a small local patch of the object at a time and many measurements have to be combined to generate a full model of an object [4, 7]. Two problems are caused by the need to have many measurements: Firstly, it is time consuming and secondly combining several local measurements in a realistic scenario where the object may move when touched is difficult.

Already in 1984, Allen presented the powerful idea of integrating vision and touch to generate surface descriptions [1]. This was later extended in [2] and [19]. An object model was initialized as surface patches from stereo vision and tactile

exploration was then used to further improve the model. The problems of the approach are similar to pure tactile exploration, that is, many probings are needed and the object must not move. The approach presented in this paper differs from these in two important respects: A single grasp action is used to collect tactile information and the object motion during exploration is explicitly included in the model so that it does not present a problem for the estimation. To our knowledge, there have been no articles during the last decade presenting methods for 3-D object model reconstruction based on combining visual and tactile data.

III. 3-D OBJECT RECONSTRUCTION

This section presents the proposed method for object reconstruction, which fuses visual and tactile information in an optimal estimation framework. First, the reconstruction problem and the object symmetry model are described, followed by the state estimation approach using Iterated Extended Kalman Filter (IEKF). Next, the estimation process using vision is given, followed by the tactile estimation model.

A. Problem description

The proposed reconstruction approach is based on a system which includes a camera capable of producing 3-D data (a stereo camera or for example Kinect) and a robotic arm equipped with a gripper including tactile sensors. A 3-D model of a symmetric object is created first using the point cloud of the object and a symmetry assumption. This model is used to plan an initial grasp which is executed. When the grasp is closed, the tactile information is used to update the 3-D model. The main challenge is how to optimally combine visual and tactile information, which is solved using IEKF which allows to minimize the joint uncertainty of the estimation taking into account uncertainties of both visual and tactile measurements.

The object is modeled using a point cloud representing the visible front part of the object. Additionally, the symmetry of the object is modeled using a symmetry plane for which the location and orientation are estimated. Finally, the location of the object (point cloud) in the robot coordinate frame, which we call *bias*, is also estimated, because of uncertainty in the robot-camera calibration and possible motion of the object while grasping it. All the parameters are initially estimated using vision and then refined using tactile measurements in an optimal estimation framework. The object is assumed to lie on a supporting plane parallel to XY-plane of the robot coordinate system. The symmetry is assumed to be along a plane oriented along the Z-axis. An example 2-D top view can be seen in Fig. 1.

The unknown state, $\mathbf{x}$, consist of bias and symmetry parameters and location of every point in the point cloud model of the object,

$$\mathbf{x} = (\mathbf{b}, \mathbf{m}, \mathbf{p}_1, ..., \mathbf{p}_N)^T, \tag{1}$$

where there are N points $\mathbf{p} = (x, y, z)^T$ in the point cloud model. The bias $\mathbf{b}$ is represented using four parameters, rotation b_θ around Z-axis and translation $\mathbf{b_T} = (b_x, b_y, b_z)^T$ in XYZ coordinates. The mirror symmetry $\mathbf{m} = (m_\theta, m_r)^T$ is represented using two parameters, rotation m_θ and distance from the origin m_r.

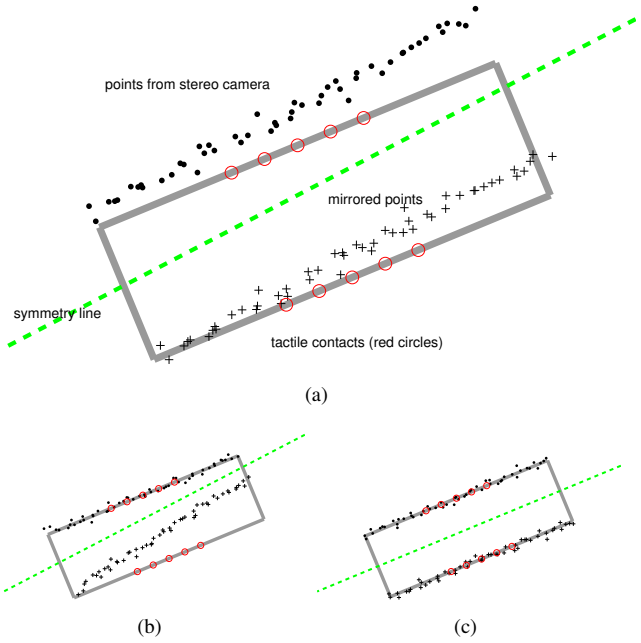


Fig. 1. 2-D visualization of estimated parameters when grasping a rectangular object (gray box); (a) the original state; (b) bias between camera and robot frames has been corrected; (c) the symmetry line has been positioned correctly.

Applying the bias to a point $\mathbf{p}$ is defined as

$$\mathbf{p}_b = f_b(\mathbf{p}, b_\theta, \mathbf{b_T}) = \mathbf{R}_z(b_\theta)\mathbf{p} + \mathbf{b_T}, \tag{2}$$

where $\mathbf{R}_z(\cdot)$ is rotation around Z-axis.

The assumption that the object lies on a supporting plane and that it is symmetric along a plane parallel to the Z-axis means that the symmetry plane can be thought of as a symmetry line in XY-plane. The symmetry line can then be defined using two parameters, rotation m_θ and distance from the origin m_r, and the line equation becomes $xcos(m_\theta) + ysin(m_\theta) = m_r$.

Applying the symmetry line requires mirroring the original point across the line. For point $\mathbf{p}$ the mirrored point is

$$\begin{aligned}\mathbf{p}_m =& f_m(\mathbf{p}, m_r, m_\theta) \\ =& \begin{bmatrix} xcos(2m_\theta) + ysin(2m_\theta) - 2rsin(m_\theta) \\ xsin(2m_\theta) + ycos(2m_\theta) + 2rcos(m_\theta) \\ z \end{bmatrix}.\end{aligned} \tag{3}$$

For the front part of the point cloud, facing the camera, only bias parameters are used but for the symmetric mirrored back part also biasing is needed after mirroring by applying (3) to (2),

$$\mathbf{p}_{m+b} = f_{m+b}(\mathbf{p}, \mathbf{b}, \mathbf{m}) = f_b(f_m(\mathbf{p}, \mathbf{m}), \mathbf{b}) \tag{4}$$

where $\mathbf{b}$ and $\mathbf{m}$ denote the bias and mirroring parameters.

B. Estimation with IEKF

The measurement model relating the tactile input to the state is non-linear. Therefore, the standard Kalman filter is not suitable and Iterated Extended Kalman filter (IEKF) is used [15]. IEKF is favored over the more common EKF because of the strong nonlinearity of the measurement model, where the iterative nature of IEKF gives superior convergence. The initial state $\mathbf{x}_{init}$ and its associated uncertainty $\mathbf{P}_{init}$ is based on the visual measurements, as described in Sec. III-C.

IEKF is initialized as

$$\begin{aligned} \mathbf{x}_0^+ &= \mathbf{x}_{init} \\ \mathbf{P}_0^+ &= \mathbf{P}_{init}. \end{aligned} \tag{5}$$

At the beginning of every round of iterations, $k = 1, 2, \ldots$, of the filter a new grasp is performed and new tactile measurement acquired. The uncertainty of the bias term is increased because the object may move. If the motion can be measured, this measurement can be accomodated through prediction $\mathbf{u}_{k-1}$. However, in this paper we do not make such measurement, and therefore the dynamic model assumes that the prediction step is zero-mean,

$$\begin{aligned} \mathbf{x}_{k,0}^+ &= \mathbf{x}_{k-1}^+ + \mathbf{u}_{k-1} \\ \mathbf{P}_{k,0}^+ &= \mathbf{P}_{k-1}^+ + \mathbf{Q}_{k-1}. \end{aligned} \tag{6}$$

where $\mathbf{Q}_{k-1}$ is the covariance of the prediction. Because of the above dynamic model, $\mathbf{u}_{k-1} = \mathbf{0}$ for us, and only the covariance is updated.

IEKF then performs the estimation by iterating the following equations for $i = 0, 1, \ldots$ until convergence:

$$\begin{aligned} \mathbf{H}_{k,i} &= \left. \frac{\partial \tilde{\mathbf{T}}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k,i}+} \\ \mathbf{K}_{k,i} &= \mathbf{P}_{k,0}\mathbf{H}_{k,i}^T \left(\mathbf{H}_{k,i}\mathbf{P}_{k,0}\mathbf{H}_{k,i}^T + \mathbf{R} \right)^{-1} \\ \mathbf{P}_{k,i+1}^+ &= \left(\mathbf{I} - \mathbf{K}_{k,i}\mathbf{H}_{k,i} \right) \mathbf{P}_{k,0} \\ \mathbf{x}_{k,i+1}^+ &= \mathbf{x}_{k,0} + \mathbf{K}_{k,i} \left(\mathbf{y} - \tilde{\mathbf{T}}(\mathbf{x}_{k,i}^+) - \mathbf{H}_{k,i}(\mathbf{x}_{k,0} - \mathbf{x}_{k,i}^+) \right) \end{aligned} \tag{7}$$

where $\tilde{\mathbf{T}}(\mathbf{x})$ is the measurement model for the tactile sensors, described in Sec. III-D. In the experiments of this paper only a single grasp is performed, $k = 1$, but several iterations of the IEKF are performed to reach corvergence, $i = 1, 2, \ldots$.

C. Initial estimation with vision

The state $\mathbf{x}$ includes bias and mirroring parameters and locations of (point cloud) points described in the robot frame. To initialize these, a point cloud of the target object is generated either with a stereo camera and a standard stereo reconstruction method or with for example Kinect. The point cloud is then segmented using method presented in [13] so that only points belonging to the object are left, which is relatively trivial when the object resides on a supporting plane without touching other objects. All further processing is performed in the robot frame and therefore the point cloud, XYZ coordinates of the points belonging to the object, is transformed to the robot frame before the following steps.

Initialization of the point locations in the state is straightforward as the point cloud points transformed from the camera to the robot frame can be directly used. To make a grasp plan an initial guess must be made for the mirroring plane and those parameters can also be used in the initial state. Determining the initial mirroring state parameters is shortly explained in the experiments, Sec. IV-B. Bias includes the error caused by the transform from camera to robot frame, which can be assumed to be zero-mean, and the movement of the object during grasping, which if noticed can be included, for example by discrepancy between where the grasp plan predicts the contacts and where the contacts actually happen.

For the state covariance $\mathbf{P}$ initialization is more tricky as the proportions of the variances affect the performance of the state estimation greatly. For example, having a large initial variance (high uncertainty) for individual points compared to bias and mirroring parameters means that the individual points will be moved instead of correcting the bias and mirroring parameters. The used camera robot calibration method [9] gives a backpropagated covariance for the robot-camera pose based on the variance of detected marker positions, which can be included in the bias covariance. However, bias parameters also include the object movement during grasping, which is likely to be larger than the uncertainty of camera-robot pose. The covariances for these parameters were determined experimentally.

D. Tactile measurements

The tactile measurement

$$\mathbf{T} = (\mathbf{T}_1, ..., \mathbf{T}_M)^T \tag{8}$$

consists of XYZ locations $\mathbf{T}_i = (T_{ix}, T_{iy}, T_{iz})^T$ of the M tactile sensor elements with non-zero tactile measurements, i.e., every tactile element which is in contact with the object. The location is obtained in the robot base frame using forward kinematics. The measurement model $\tilde{\mathbf{T}}(\mathbf{x})$ then predicts the measurements based on the state. In addition to the locations of the tactile sensor elements, also their normal vectors are known, but these are only indirectly used as shown below. In the following only one tactile element $\mathbf{T}$ without index is used to simplify notation.

The measurement model is based on the idea that each tactile measurement results from a single corresponding point in the point cloud. Thus, the correspondence needs to be established. A simple approach would be to determine maximally likely correspondences, resulting in an iterative closest point (ICP) type approach. This could be done in a probabilistic framework, taking into account the uncertainty of the current parameters in the measurement model, thus giving a prediction with both location $\mathbf{T}$ and its covariance $\mathbf{C}$, as described in Sec. III-E. Now, the probability that point $\mathbf{p}_j \sim \mathcal{N}(\mathbf{T}, \mathbf{C})$ belongs to the distribution defined by $\mathbf{T}$ and $\mathbf{C}$ is given by probability $p(\mathbf{p}_j|\mathbf{T}, \mathbf{C})$. Thus, we could find the point j for which the probability would be maximal. However, this would result in each tactile measurement only affecting the location of a single point in the point cloud, which would be

problematic because there are typically many points for which the likelihood is approximately equal due to the uncertainty of mirror and bias parameters.

Thus, instead of ICP, we consider an Expectation Maximization (EM) inspired approach, where the correspondence is taken to be an unknown random variable and the estimation is based on maximization of the expected value rather than the maximum likelihood correspondence. As already mentioned, the conditional probability that a point corresponds to a particular tactile measurement is $p(\mathbf{p}_j|\mathbf{T},\mathbf{C})$, which can be understood as a weight. Moreover, to avoid infinite support, small weights will be zeroed. To have a statistically motivated cutoff threshold, the squared Mahalanobis distance, $D^2(\mathbf{p}_j) = (\mathbf{p}_j - \mathbf{T})^T\mathbf{C}^{-1}(\mathbf{p}_j - \mathbf{T})$, is calculated. When the data is normally distributed, the squared Mahalanobis distance follows χ^2 distribution with n degrees of freedom, χ^2_n, in our case with three dimensional coordinates $n = 3$. Therefore, a limit for the squared Mahalanobis distance can be established which includes certain part of the whole distribution. In this case limit which includes 95% was selected which gives the limit $D^2 < 7.815$. The weights are then defined as

$$w_j(\mathbf{p}_j) = \begin{cases} p(\mathbf{p}_j|\mathbf{T},\mathbf{C}) & \text{if } D^2(\mathbf{p}_j,\mathbf{T},\mathbf{C}) < 7.815 \\ 0 & elsewhere \end{cases} \quad (9)$$

Following the EM idea, these weights (conditional probabilities), normalized such that their sum is one represent the probabilities of points corresponding to a particular measurement and thus, the expected measurement for frontal (biased) points is equal to

$$E[\tilde{\mathbf{T}}_b] = \frac{1}{\sum_{j=1}^{N} w_{b,j}} \sum_{j=1}^{n} w_{b,j}\mathbf{p}_{b,j}, \quad (10)$$

where point $\mathbf{p}_j$ after applying the bias is called $\mathbf{p}_{b,j}$, and its weight $w_{b,j}$. The predicted location of the tactile element is thus formed as a weighted sum of biased point locations. Similarly for non-visible (back) points the measurement model is

$$E[\tilde{\mathbf{T}}_{m+b}] = \frac{1}{\sum_{j=1}^{N} w_{m+b,j}} \sum_{j=1}^{n} w_{m+b,j}\mathbf{p}_{m+b,j}. \quad (11)$$

where $\mathbf{p}_{m+b,j}$ is the point $\mathbf{p}$ after mirroring and biasing, and $w_{m+b,j}$ its weight.

The uncertainty of prediction of a tactile measurements varies depending on the density of the point cloud close to the measurement. This should be somehow taken into account in the measurement model. One way would be to change the measurement uncertainty $\mathbf{R}$ inversely proportionally to the sum of weights, so that less weight means more uncertainty. Here, the same effect has been achieved by scaling the residuals according to

$$a_b = \frac{\sum_{j=1}^{N} w_{b,j}}{a_{MAX}} \text{ and } a_{m+b} = \frac{\sum_{j=1}^{N} w_{m+b,j}}{a_{MAX}}, \quad (12)$$

where a_{MAX} is the maximum a_b or a_{m+b} over all tactile elements. The goal during the state estimation is then to minimize the residual of $a_b(\mathbf{T} - \tilde{\mathbf{T}}_b)$ and $a_{m+b}(\mathbf{T} - \tilde{\mathbf{T}}_{m+b})$. This gives the tactile elements close to many points more importance than those near solitary points.

For Kalman filter update, the Jacobians of the measurement models $\frac{\partial\tilde{\mathbf{T}}_b}{\partial\mathbf{x}}$ and $\frac{\partial\tilde{\mathbf{T}}_{m+b}}{\partial\mathbf{x}}$, are needed, that is, the partial derivatives with respect to all variables in the state. For a first order approximation, the effect of derivative of the weight is assumed to be negligible compared to the derivative of (2) and (4) and therefore the weight is not assumed to change. Therefore the partial derivative for $\frac{\partial\tilde{\mathbf{T}}_b}{\partial\mathbf{x}}$ is calculated as

$$\frac{\partial\tilde{\mathbf{T}}_b}{\partial\mathbf{x}} = \frac{1}{\sum_{j=1}^{N} w_{b,j}} \sum_{j=1}^{n} w_{b,j}\frac{\partial\mathbf{p}_{b,j}}{\partial\mathbf{x}} \quad (13)$$

and similarly for $\frac{\partial\tilde{\mathbf{T}}_{m+b}}{\partial\mathbf{x}}$. Note that the Jacobians also depend on the individual point locations and therefore the point locations will be adjusted during state estimation.

The measurement error covariance $\mathbf{R}$ is initialized as a flattened ellipsoid in the direction of normal vector of each tactile element. The objective is that the state estimation should be more inclined to fix errors in the direction of the normal than along the plane of the tactile sensor.

E. Propagation of measurement covariance

The measurement covariance $\mathbf{C}$, used to determine correspondence, depends on two independent components, one representing the uncertainty of contact location within the tactile sensor and another representing the uncertainty in the sensor location due to the uncertainty in the current state $\mathbf{P}$. The fact that covariance matrices can be combined by simple addition is utilized.

The constant component $\mathbf{C}_{tact}$ is formed as an elongated ellipsoid which points to the direction of the normal of the tactile element. A tactile element "sees" further in the direction its normal points to, but not far in the plane of the tactile element array to avoid overlap between neighboring tactile elements.

The second component of the uncertainty depends on the current state uncertainty, $\mathbf{P}$. The state includes four bias parameters, $\mathbf{b} = (b_x, b_y, b_z, b_\theta)$, and two mirroring parameters, $\mathbf{m} = (m_r, m_\theta)$. Their covariance must be propagated to the world coordinates. In general given a non-linear function $f : \mathbb{R}^M \to \mathbb{R}^N$ and a random vector $\mathbf{v}$ in $\mathbb{R}^M$, the approximation of mean and covariance of $f(\mathbf{v})$ can be computed in the vicinity of the mean $\overline{\mathbf{v}}$ of the distribution. The approximation of f is $f(\mathbf{v}) \approx f(\overline{\mathbf{v}}) + \mathbf{J}_f(\mathbf{v} - \overline{\mathbf{v}})$, where $\mathbf{J}_f$ is the Jacobian $\frac{\partial f}{\partial\mathbf{v}}$ evaluated at $\overline{\mathbf{v}}$. The first-order approximation of random variable $f(\mathbf{v})$ has mean $f(\overline{\mathbf{v}})$ and covariance $\mathbf{\Sigma}_f = \mathbf{J}_f\mathbf{\Sigma}\mathbf{J}_f^T$ [8].

The covariance of the bias parameters in the state covariance, $\mathbf{C}_{P,b}$, propagated to the world coordinates using Jacobian of (2) evaluated at the position of the current state is

$$\mathbf{C}_b = \left(\frac{\partial f_b}{\partial\mathbf{b}}\bigg|_x\right)\mathbf{C}_{P,b}\left(\frac{\partial f_b}{\partial\mathbf{b}}\bigg|_x\right)^T \quad (14)$$

and similarly for biased+mirrored points with state covariance $\mathbf{C}_{P,m+b}$ with Jacobian of (4)

$$\mathbf{C}_{m+b} = \left(\frac{\partial f_{m+b}}{\partial \mathbf{b}, \mathbf{m}} \bigg|_x \right) \mathbf{C}_{P,m+b} \left(\frac{\partial f_{m+b}}{\partial \mathbf{b}, \mathbf{m}} \bigg|_x \right)^T. \quad (15)$$

The covariance $\mathbf{C}_{tact} + \mathbf{C}_b$ and the tactile element location $\mathbf{T}$ are then applied in (9) to calculate $w_{b,j}$ for biased points and similarly with covariance $\mathbf{C}_{tact} + \mathbf{C}_{m+b}$ for mirrored+biased points to calculate $w_{m+b,j}$.

IV. EXPERIMENTS

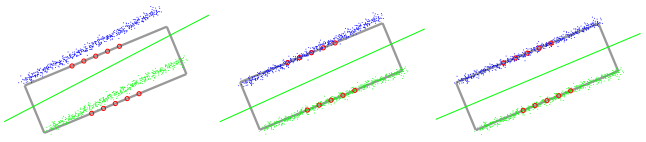
The following experiments demonstrate the performance of the method. We begin with experiments using generated 2-D and 3-D data and follow with experiments using point clouds captured with Kinect and objects grasped with a real robot. There are no comparisons to other similar methods, because the authors are not aware of existence of other similar methods combining visual and tactile data.

Note that while the method can and will adjust both bias/mirroring parameters and individual point locations at the same time, it is important that the bias and mirror parameters are approximately corrected first or the errors may be reduced by moving the points unrealistically. This is because large initial errors in bias and mirror parameters will cause gross correspondence errors. Therefore, the state estimation proceeds by initially adjusting only bias/mirror parameters. After a number of IEKF iterations, also point locations are included in the state.

A. Generated data

We begin with a simple 2-D example which illustrates the operation of the system. Then, we present a 3-D experiment demonstrating a more complex scenario with multiple contact surfaces.

1) 2-D example: The first example is similar to Fig. 1 where the bias and mirror parameters are initially incorrect. Gaussian noise has been used to corrupt the point cloud point locations. The results can be seen in Fig. 2 for initial estimate, and after 4 and 10 IEKF iterations. Even though the initial parameters are significantly wrong, only ten iterations of the IEKF loop were able to correct them. The method is able to correct also the individual point locations, however, in this case the effect is not easily visible because there were only 5 tactile points on each side.



(a) initial (b) $i = 4$ (c) $i = 10$

Fig. 2. A 2-D example of adjusting bias and mirroring plane, where blue (upper) points represent the visible part of the object, green points the non-visible back and red circles the tactile points.

2) 3-D example: In this experiment, the front of a cylinder is visible and tactile elements in three distinct locations ("fingers") are used. The experiment demonstrates the ability of the proposed method to use any number of tactile elements and that the tactile elements can be in any configuration. Initial bias and mirroring parameters were set off, as can be seen in Fig. 3(a). After 30 iterations the bias and mirroring parameters were approximately corrected, Fig. 3(b), after which also the point locations were included in the state updates and the state converged at iteration 415 as shown in Fig. 3(c–d). A detail of point movement near the rightmost tactile element can be seen in Fig. 3(e), where the point motion is shown as lines. The point trajectories do not point strictly towards the tactile points, because the state was updated jointly and bias and mirroring parameters were updated at the same time as the point locations.

B. Real data

1) Set up: Experiments with real data were performed using Mitsubishi Melfa RV3-SB industrial robot arm, Schunk PG-70 parallel jaw gripper equipped with tactile sensors made by Weiss robotics and Kinect. The gripper and type of the tactile sensor can be seen in Fig. 4.

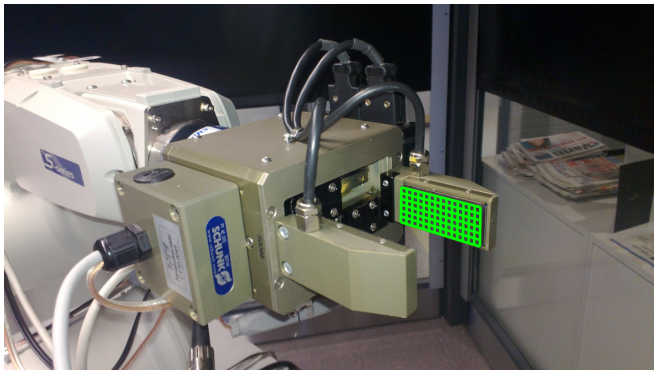
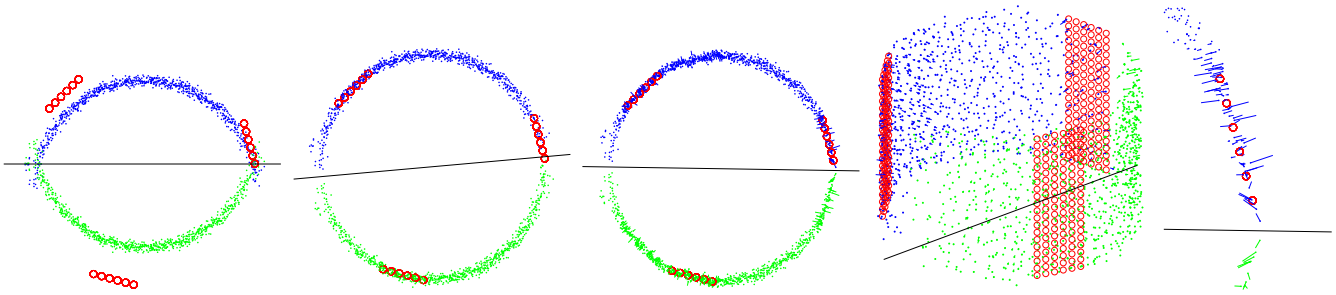


Fig. 4. Gripper equipped with tactile sensors. The green grid highlights the tactile element with 6×14 tactile "pixels" and measuring $22 \times 50mm$.

Robot-camera calibration was performed using the method presented in [9]. The method uses a LED marker attached to the gripper and automatically finds the pose between camera and robot frames in axis-angle and translation form. The method also provides uncertainty of the transformation based on the backpropagated variance of the marker locations. How much do we say about Kinect calibration? Nothing is the easy way...

After the calibration was performed, an object was placed on a table, a point cloud was captured using Kinect and the point cloud was segmented. Point cloud segmentation is an important and in cluttered scenes difficult task, but as the main interest here lies elsewhere the process was simplified by having the single object of interest alone in an otherwise empty table. The pointcloud was segmented using the method presented in [13] and the the largest segment in the robot's working area was selected as the object of interest.



(a) initial (b) $i = 30$ (c) $i = 415$ (d) $i = 415$ in 3-D (e) $i = 415$ detail

Fig. 3. A 3-D example; (a) Initial setting; (b) 30 iterations of IEKF corrected bias and mirroring; (c–d) at iteration 415 also the point locations were straightened near tactile elements; (e) a detail near the rightmost tactile element.

2) Grasp plan and parameter initialization: A grasp plan must be generated to grasp the selected object and the grasp plan should also provide initial parameters for the location of the mirroring plane. The method presented in [5] could be used for both of these tasks, but a simpler approach was chosen as the main interest here was combining visual and tactile data, not grasping itself. The applied method first finds the principal components using PCA in the point cloud segment. The first component and Z-axis (direction of gravity) define the direction of the mirroring plane. The center point for the grasp and estimate for object width is determined based on the distribution of the data along vector orthogonal to the mirroring plane.

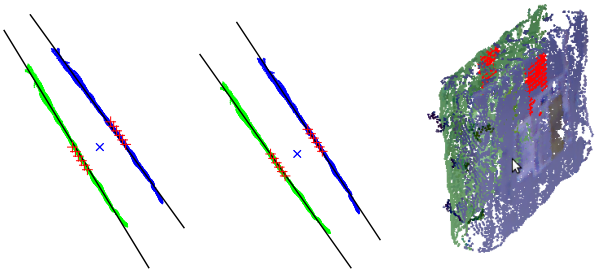
The gripper was then moved to grasp the object orthogonally to the mirroring plane directly from above using the widest possible jaw opening and the gripper was closed until sufficient tactile contact was measured. The mirroring plane was initialized based on the first principal component and the center point of grasp, and the bias parameters based on the estimated width of the object and the realized gripper width, i.e., how far the front part of the point cloud was compared to where it was assumed to be. Later in the results this initial stage is called as Stage 0. If the direction of the mirroring plane was determined correctly by the grasp planner and there was no bias (no calibration error and no object movement) the result is already perfect because the object width is implicitly included by setting the mirroring plane to the middle point of the grasp. During the next stage, Stage 1, the mirroring and bias parameters were adjusted and the final Stage 2 adjusted also the individual point locations.

Due to limitations of the used hardware (parallel jaw gripper with maximum aperture of 65mm), the set of used objects was limited to relatively simple and thin objects. Also some of the strengths of the fusion method cannot be displayed with parallel jaw configuration, because of limited tactile information. Nevertheless, four objects were chosen, Fig. 5: a CD-drive, a salt container (cylinder), a plastic case (housing a jigsaw puzzle) and a spray bottle. For the first two objects quantitative results were measured and qualitative observations were made for the last two.

Fig. 5. The test objects.

3) Numerical results: First results are for the CD-drive. Quantitative measurements were performed after the sensor fusion and the measurements in this case were the thickness of the drive, measured as $42mm$, and the angle between the two sides, which obviously should be $0°$. Both of the measurements were based on robustly fitting lines (using `robustfit()` function in Matlab) to a top-down (projected to XY-plane) view of the front and back parts of the point cloud. The width was measured as the shortest distance from the center of grasp. A visual example of the results can be seen in Fig. 6. The actual results are in Table I. The results improved with further stages of the fusion process except in few cases. For example angle in measurent 1, where the initial angle was measured almost perfectly at $0°$, was worsened slightly because of the small width of the sensor ($22mm$) and local imperfections in the generated point cloud. Majority of the average error for the angle came from a single measurement, where the large initial error in mirroring parameters was not corrected completely.

Next results are with the salt container, which is a cylinder with diameter $53mm$. In this case the error was measured as the diameter of the point cloud projected to the XY-plane. The circle fit was performed using an implementation of Taubin's circle fit algorithm [17]. Visual example of the results is in Fig. 7 and the results in Table II. The results improved in all cases. The constant slight overestimation of the diameter was caused by fitting a planar representation of a tactile sensor to a



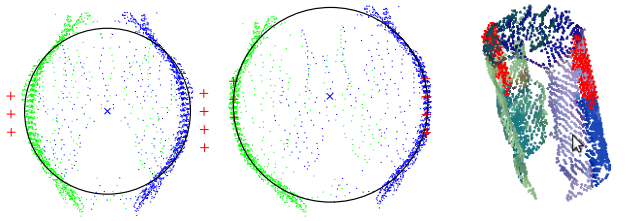
(a) Stage 0 (b) Stage 2 (c) Reconstruction

Fig. 6. Results with the CD-drive (measurement 2); (a&b) Top-view; blue is the front and green is the mirrored back part with fitted black lines. Cross marks the center of the grasp, red plus-marks the tactile points; (c) Reconstructed object with the same color coding.

TABLE I
RESULTS WITH A CD-DRIVE, THE MEASURED THICKESS WAS $42mm$ AND THE ANGLE BETWEEN OPPOSING SIDES $0°$.

	Thickness (mm)			Angle (deg.)		
Stage	0	1	2	0	1	2
1	39.35	40.08	39.55	0.79	6.88	4.96
2	38.85	41.71	41.65	4.47	4.89	1.90
3	36.10	44.00	43.35	7.53	0.99	3.62
4	31.37	39.43	39.83	13.90	9.10	9.19
5	40.40	41.82	41.84	0.33	0.70	0.64
mean	37.22	41.41	41.24	5.40	4.51	4.06
std.dev.	3.63	1.78	1.57	5.58	3.67	3.30

cylindrical pointcloud, i.e., the middle part of the tactile sensor ”sinks” into the cylinder while in reality the tactile sensor gets deformed and in fact is not a plane when grasping a cylinder.



(a) Stage 0 (b) Stage 2 (c) Reconstruction

Fig. 7. Results with the salt container (measurement 1).

TABLE II
RESULTS WITH A SALT CYLINDER, DIAMETER $53mm$.

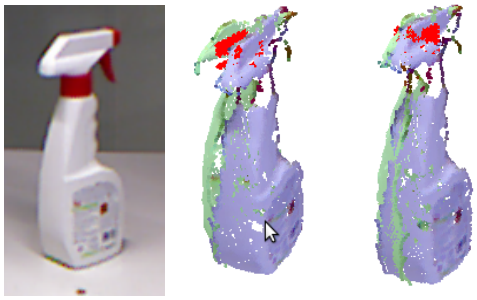
	Diameter (mm)		
Stage	0	1	2
1	45.85	53.96	54.08
2	43.14	57.13	56.35
3	45.53	53.99	53.94
4	45.48	54.28	54.31
5	47.68	55.58	55.57
mean	45.54	54.99	54.85
std.dev.	1.62	1.37	1.06

4) Qualitative results: Some qualitative observations with the remaining two objects follow. The plastic case is a nearly rectangular object, but with rounded corners and a handle at the top. In general the reconstruction succeeded well, but the problems noticed with the CD-drive were slightly more common. Only the tip of the tactile sensor hits the sides of the case because the handle prevents a close grasp. Therefore, local disturbances in the pointcloud caused more problems because there was tactile information available only from a small area. A successful reconstruction example can be see in Fig. 8.

(a) (b)

Fig. 8. The plastic case; (a) the original; (b) reconstruction after grasp.

The last object, a spray bottle, is more complex than the other objects and determining the symmetry plane succeeded well only when the side of the bottle was facing almost directly to the camera. Because of this the grasp was often done from a wrong angle and the object was rotated even tens of degrees when grasped. This caused the initial symmetry parameters to be off by a large margin. An example of this can be seen in Fig. 9. In the initial reconstruction there was a wide gap in the back of the bottle, but this was corrected in the final reconstruction.



(a) (b) (c)

Fig. 9. Results with the spray bottle; (a) the original; (b) initial reconstruction; (c) final reconstruction.

V. CONCLUSION

In this work a sensor fusion method combining visual and tactile information was presented. The visual model is captured as a 3-D point cloud and after grasping the tactile information is combined with visual data applying a state estimation method, iterated extended Kalman filter (IEKF). The method was validated in experiments with artificially generated data and in real experiments where visual data was captured with Kinect and tactile data with a robotic arm.

There is still potential for further work. The most immediate addition is to perform experiments using a more complex robotic hand and to compare the grasp planner and visual symmetry estimation method in [5] to the ones proposed here. Also more comprehensive quantitative results comparing the created object models to ground truth, created by laser scanning the objects, are planned. In the longer term a more comprehensive system for grasping can be created, where the model is updated before trying to manipulate the object and the stability of the grasp is evaluated based on the tactile information before lifting the object [3]. Then, if a grasp appears to be unstable, another grasp plan can be created using the improved model.

REFERENCES

[1] P. Allen. Surface descriptions from vision and touch. In *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, volume 1, pages 394 – 397, mar 1984. doi: 10.1109/ROBOT.1984.1087191.

[2] P.K. Allen. Integrating vision and touch for object recognition tasks. *The International Journal of Robotics Research*, 7(6):15–33, 1988.

[3] Yasemin Bekiroglu, Janne Laaksonen, Jimmy Jorgensen, Ville Kyrki, and Danica Kragic. Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616–629, 2011.

[4] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A potential field approach to dexterous tactile exploration of unknown objects. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 360 –366, dec. 2008. doi: 10.1109/ICHR.2008.4756005.

[5] Jeannette Bohg, Matthew Johnson-Roberson, Beatriz Leon, Javier Felip, Xavi Gratal, Niklas Bergstrom, Danica Kragic, and Antonio Morales. Mind the gap - robotic grasping under incomplete observation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 686 –693, may 2011. doi: 10.1109/ICRA.2011.5980354.

[6] C. Colombo, A. Del Bimbo, and F. Pernici. Metric 3d reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1):99 –114, jan. 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.14.

[7] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. Gaussian process implicit surfaces for shape estimation and grasping. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2845 –2850, may 2011. doi: 10.1109/ICRA.2011.5980395.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd. edition*. Cambridge University Press, 2003.

[9] J. Ilonen and V. Kyrki. Robust robot-camera calibration. In *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*, pages 67–74, Estonia, 2011.

[10] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3304 –3311, may 2011. doi: 10.1109/ICRA.2011.5980145.

[11] D.C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2136 –2143, june 2009. doi: 10.1109/CVPR.2009.5206872.

[12] M. Popovi, G. Kootstra, J. A. Jrgensen, D. Kragic, and N. Krger. Grasping unknown objects using an early cognitive vision system for general scene understanding. In *International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[13] M. Richtsfeld and M. Vincze. Grasping of unknown objects from a table top. In *ECCV 2008 Workshop on 'Vision in Action: Efficient strategies for cognitive agents in complex environments'*, 2008.

[14] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173, 2008.

[15] Dan Simon. *Optimal State Estimation*. John Wiley & Sons, 2006.

[16] J. Speth, A. Morales, and P.J. Sanz. Vision-based grasp planning of 3d objects by extending 2d contour based algorithms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2240 –2245, sept. 2008. doi: 10.1109/IROS.2008.4650632.

[17] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(11):1115 –1138, nov 1991. ISSN 0162-8828. doi: 10.1109/34.103273.

[18] Tianfan Xue, Jianzhuang Liu, and Xiaoou Tang. Symmetric piecewise planar object reconstruction from a single image. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2577 –2584, june 2011. doi: 10.1109/CVPR.2011.5995405.

[19] Y. Yamada, A. Ishiguro, and Y. Uchikawa. A method of 3d object reconstruction by fusing vision with touch using internal models with global and local deformations. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 782 –787 vol.2, may 1993. doi: 10.1109/ROBOT.1993.291939.

What do contacts tell about an object?

Ekaterina Nikandrova and Ville Kyrki

Abstract— **Among all senses the sense of touch is the only one without which humans are not more able to control and manipulate objects. Similarly, tactile sense is invaluable for robotic manipulation in uncertain environments. It is however not thoroughly understood to what extent properties of the robot environment can be inferred from the tactile sense. This paper presents a novel approach that allows to study how much information a robot can optimally learn from a single tactile exploration attempt. Our method makes use of a simulator as an internal memory for the robot. The evaluation is based on assessing how much information error minimization between predicted and actual sensor readings can provide about the environment. This paper focuses on evaluating geometric parameters in a transportation task. Experiments performed with a set of objects with various shapes indicate that a single exploration action is not guaranteed to provide much information for all uncertain factors if the attempt is not originally planned for information gain in mind. Moreover, the information gain for different attributes varies significantly depending on the object geometry.**

I. INTRODUCTION

Touch is one of the five senses through which animals and people interpret the world around them. It is practically impossible to hold or safely manipulate various object without touching them. As in humans, touch sensing in robotics could help in understanding the interaction with real world objects. The sense of touch is particularly important in manipulation as it allows to estimate properties, such as geometry, stiffness, and surface condition. The importance of touch in manipulation is also underlined by the fact that humans rely mostly on the sense of touch when performing manipulation tasks [1].

The sense of touch has a wide variety of uses in robotics. Initially, studies in tactile sensing area focused on the development of sensor devices and object recognition algorithms [2]. Over the past years, tactile sensors have been applied in numerous tasks, including object classification and recognition [3], [4], pose estimation [5], as well as grasping [6]. However, despite the great number of recent works in the effective use of tactile and contact information in solving robotics problems, it is not thoroughly understood to what extent properties of the robot environment can be inferred from the tactile sense.

Our goal in this paper is to study to what degree a robot can use the tactile sense to learn about its environment. We propose a novel simulation-based approach that provides a possibility to evaluate the amount of information about the object that can be obtained from a single tactile exploration attempt. The term "information" in our case implies an evaluation of different object parameters.The distinguishing feature of our approach is to use simulation as an internal model of the environment for the robot. Thus, the simulator plays the role of the memory for the robot and allows the robot to try out actions before executing them for real. More than that, simulation provides precise results, without measurement uncertainty, which allows us to study the question "how much can be done with a certain type of sensors" rather than just "how good sensors are used".

The approach is based on minimizing the difference between predicted and measured sensor readings. This difference indicates the error in the robot's expectation of the environment compared to the true state of the environment. By minimizing this error by modifying the initial conditions of the simulation, the system can update its internal view of the environment. To study our research question about the limits of tactile sensing, we can then compare the updated internal view to a known true state of the environment, as the true state is known in simulation.

In a concrete evaluation scenario, we consider an object transportation task having errors in geometric attributes including the object pose and size. Grasping and object transportation are most common tasks in which tactile information is a key factor for successful implementation of a plan. Contact sensors which detect the presence or absence of contacts between the robot hand and a target object together with information about joint angles of the fingers after closing the hand are used as information sources. These measurements were chosen because, firstly, such data can be reliably obtained both in simulation and with a real robot. Secondly, in the exploration scenario it is not desirable to change the world by moving the object, so only small contact forces should be used.

The aim of this study is to inspect the capabilities of contact sensors in general. Thus, to reduce bias in the results due to the choice of particular models, the experiments have been performed with several object models, including bodies with simple geometry (tea or marmalade boxes) and more complex asymmetrical examples (cup, spray flask, pitcher, toy car). To perform the optimization, we studied several optimization algorithms to avoid bias in the results because of a particular algorithm performance. The algorithms used include both directional methods (Steepest descent) and metaheuristics (Simulated Annealing, Particle Swarm optimization and Firefly algorithm).

An important finding that should be underlined is that the inference is surprisingly difficult from a single explorative action. The level of difficulty of assessing different object attributes varies a lot. For example, it is much more difficult to determine a real object's orientation than its position or size. One reason for such complexity is the presence

Preprint submitted to Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics. Received January 27, 2012.

of multiple local minima in the fitness function. Another explanation is the ambiguity problem, that is, the global minimum is not unique but instead there is a manifold in which the obtained tactile information is the same. The importance of this fact is that no method will ever be able to disambiguate the optimum without further information.

II. RELATED WORK

Early works using tactile sensors were limited to industrial robotics [2]. After the creation of multifingered robot hands in 1980's, the use of the sensors extended to control of manipulation [7]. Most recent studies in tactile perception are focused on object classification, recognition and localization problems. Previously, tactile sensors were used to explore 3D object shapes by collecting a point cloud to constraint an object geometry [9] or by creating volumetric models [10]. Alternatively, recent works try to build an object model directly from haptic sensor data without building a 3D model [11]. Last years there are appeared several works based on novel "bag-of-features" methods successfully applied in vision-based object-recognition systems and adapted to haptics [12],[13]. Moreover, current applications include control of manipulation [8], pose estimation [14], as well as estimating as well as estimating the state of the object [15] or the interaction, such as grasp stability [16].

Our work relates mostly to the pose estimation, although we consider the scale of the object in addition. An influence of tactile measurements on localization error has been studied, primarily, for the industrial workpiece localization problem [29], [30]. Object localization using tactile sensors is based on one of two basic ideas: either, the estimation is performed by minimizing a cost function to find a solution which best fits the measurements [17], [18], [19]. Alternatively, Bayesian state estimation can be used to capture different types of uncertainties in the estimation process [21]. In the most recent research by Petrovskaya proposed an efficient Bayesian approach termed Scaling Series for global object localization via touch. This is a Monte-Carlo approach, that performs a series if successful refinements in combination with annealing [20].

The approach presented in this paper is based on optimization, because our main goal is to study the problem in an ideal environment without measurement uncertainty. This is made possible by using simulation to provide noise-free measurements, allowing us to focus on studying how much can be inferred using the tactile sensors in the ideal case.

Nowadays, simulation plays an important role in robotics and especially in robotic manipulation, for example to perform grasp planning [22],[16]. Simulation of tactile sensors is applied in many studies but in contrast to implemented techniques our approach uses only contact information.

III. LEARNING THROUGH SIMULATION

A. General approach

The initial motivation for the present work was to try to determine how much information about an object can be obtained from a single grasp attempt by using the simulator as a "memory" for a robot, which allows the robot to try out actions in simulation before the real execution. To ensure that the robot will succeed with its plan the simulation model should be updated based on error minimization results.

Our approach is based on a scenario, where a robot will first plan an action based on its current world knowledge. Then, the robot performs the action collecting sensor data during the attempt. After the action, the robot world model is updated so that the collected sensor data is best explained by the world model. Finally, the action can be replanned with the updated world knowledge, if necessary. A general process structure for the scenario is schematically presented in Fig.1.

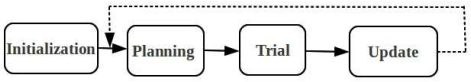


Fig. 1: General process structure

The diagram consists of four basic blocks. At the *Initialization* part the robot makes use of its internal mental view and obtains an initial guess for the object attributes (e.g. location). With use of this predicted value the robot *plans* a trajectory to complete the task. After that, this trajectory is *tried* out and actual sensor readings are obtained. In order to minimize the difference between planned and real values the *Update* block is executed. Update includes optimization procedures, which allow to find a new state for the simulation model. If one optimization step is not sufficient, for example, the action was not successful, the procedure is repeated starting with planning a new trajectory for the updated world state.

It is important to point that simulation is used in three of four blocks: *Planning*, *Trial* and *Update*. Use of a simulation for planning or trying some actions is what can be met quite often in robotics. However, application of simulation for the world model update and change of the action plan before its execution for real is a new idea, which transforms the simulation process into the internal mental view of the robot.

B. Fitness function

The choice of a fitness function can affect dramatically the method performance. The geometry of a function is a characteristic, which has direct influence on the algorithm convergence to a global minimum. A fitness function E in this study is the difference between real and predicted measurements which is determined by the sum of two components: contacts E_{cont} and finger joint angles E_{ang}. Thus, it is represented by a weighted sum of squared differences of planned and really measured collision matrices $coll$ and $coll_r$ as well as by the sum of squared differences in planned ang and measured ang_r finger angles at the moment of complete close around the graspable object. The two components are weighted to compensate for their different ranges by multiplying the joint angles part by a constant factor c_{ang}

Preprint submitted to Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics. Received January 27, 2012.

(1).

$$E = E_{cont} + E_{ang}, \text{where}$$
$$E_{cont} = \sum_t e^{\frac{-\left(t_{cur}-t_{first}\right)}{a}} \sum_i (coll^{(t,i)} - coll_r^{(t,i)})^2, \text{and}$$
$$E_{ang} = c_{ang} \sum_j (ang^j - ang_r^j)^2. \quad (1)$$

The form of the fitness function was obtained after several trials. Initially, the function included only contact information, multiplied by the exponential weight function, in which t_{cur} is a time instant when the first difference between planning and real contacts occurred, t_{first} is a current time instant and a is a damping coefficient that affects the rate of exponential decrease. It is monotonically decreasing function, which gives bigger weights to errors arisen at an earlier time. However, the analysis of the fitness function shape revealed several problems like multimodality, large number of neutral (flat) areas, deceptiveness and premature convergence to local minima. The modified fitness function, which includes additional information about finger angles, shows several improvements. It typically has only one global minimum and less flat areas. Nevertheless, the function remains deceptive in some parts of its shape.

IV. OPTIMIZATION

To update the world state, the fitness function presented in the previous section is minimized. Due to the use of simulation and complex non-linear relationships between the world state and the sensors, the minimization is not a trivial task. To begin with, the fitness function is multimodal and non-linear. Furthermore, the choice of the optimization method needs to consider factors such as accuracy and computational complexity. In order to avoid bias in our results due to the choice of a particular algorithm, we studied several algorithms including both directional methods and metaheuristics.

A. Directional methods

Directional methods are very widely used in optimization. For this reason, we chose to implement one directional method, Steepest Descent, which is a gradient-based algorithm.

In order to perform steepest descent search, gradient of the optimized function needs to be evaluated. In our case the fitness function is a result of a simulation run which makes it impossible to evaluate the gradient analytically. Instead, we chose to use a two-sided finite difference approximation, that is, for each variable x,

$$\frac{\partial f}{\partial x} = \frac{f(x+dx) - f(x-dx)}{2dx} \quad (2)$$

where dx was chosen experimentally. The partial derivative needs to be evaluated for each unknown variable, thus requiring two simulation runs for each variable. Finally, the step size λ needs to be determined. A constant step size is unlikely to provide good results, especially because the evaluation of the gradient is rather costly. Therefore, we considered the step size selection a one-dimensional optimization problem, which was solved using golden section search, as described in [23, pp. 398–400].

B. Metaheuristics

Due to several local optima of the fitness function, directional methods are not able to explore the search space outside a region around the initial local optimum. More than that, in some regions the function shows deceptivity in its behavior. This means that the gradient leads the optimizer away from the global optimum.

To cope with these problems, we also study metaheuristics approaches. They are useful for many ill-structured global optimization problems which contain several local optima or stationary points. The approaches use randomness in order to avoid getting stuck to local optima as well as avoid evaluating the gradient. Metaheuristics have been found to be able to locate a nearly optimal solutions within a reasonable computational time and use of memory without any requirements of derivatives or careful choice of initial values. We consider three different metaheuristics: Particle Swarm Optimization, Simulated Annealing, and Firefly algorithm.

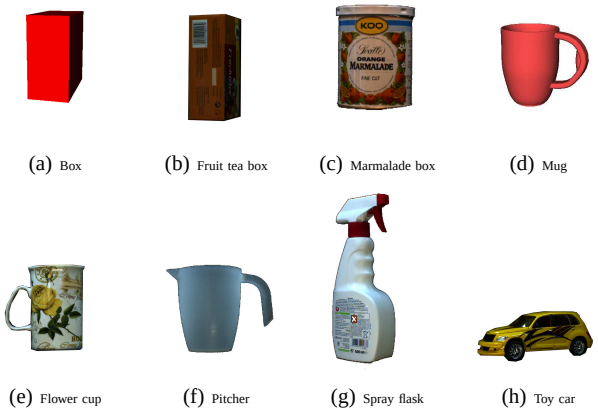
1) Particle Swarm Optimization: Particle Swarm Optimization method (PSO) was developed by Kennedy and Eberhart in 1995 [24] based on observations of behavior in a bird flock and fish school. Most important advantages of this approach are simplicity of implementation, low number of parameters and absence of derivative calculations. It is a very efficient global search algorithm, but its weak point is slow convergence in refined search stage (weak local search ability). PSO has successfully been applied to purposes such as training of neural networks, optimization of electric power distribution networks, structural optimization or system identification in biomechanics.

2) Simulated Annealing: Simulated annealing (SA) is a numerical optimization algorithm based on the metal annealing processing. First it was described by S. Kirpatrick in [25], but the idea of SA came from a paper published by Metropolis et al. [26] in 1953.

Simulated annealing is a robust and general technique which can deal with highly nonlinear models, chaotic and noisy data and many constraints. At the same time, since SA is a metaheuristic, a lot of choices are required to turn it into an actual algorithm. SA is extensively applied for combinatorial optimization problems in computer-aided design and also for image processing purposes.

3) Firefly algorithm: The Firefly Algorithm (FA) is a novel metaheuristic which was developed by Xin-She Yang [27] and is based on the flashing patterns and behavior of fireflies. The algorithm is similar to PSO in a sense that there are a lot of particles (fireflies) moving in a search space according to specific dependences. All particles are characterized by the "attractiveness" factor, which is proportional to their fitness (brightness) and they both decrease as their distance increase. If there is no particle with best fitness then a particular firefly will move randomly.

The advantage of FA compared with PSO is its ability to deal more naturally and efficiently with multimodal func-



(a) Box (b) Fruit tea box (c) Marmalade box (d) Mug

(e) Flower cup (f) Pitcher (g) Spray flask (h) Toy car

Fig. 2: Various objects used for experiments

tions. Digital image compressing, feature selection and fault detection or eigenvalue optimization are examples of FA's recent applications.

V. EXPERIMENTS

A. Design of experiments

Several detailed and more general questions were posed within this study. First and the most important general question to answer is how much can be learned about the object from a single tactile exploration action? What are the limitations of the proposed approach and how far it can be extended? Another point is to determine the set of object parameters that can be estimated and to specify the level of how well these attributes can be determined. All experiments were designed in order to solve these questions.

Four parameters divided in three categories were chosen as the set of unknown parameters. These are location group (x,y coordinates in meters), orientation (angle of rotation around z-axis in degrees) and the scale factor of object size. We performed two experiments: In Experiment 1, we consider only location and orientation as unknowns, totaling 3 unknown DOF. In Experiment 2, also the scale factor is unknown, so that there are four unknown DOF.

In order to verify that the approach is suitable in solving the given problem and to increase the generality of the results, we studied also the sensitivity of the method to object geometry variations. For testing our approach, we chose 8 objects shown in Fig. 2. The set was chosen so that it includes objects with simple geometrical shape (box, tea box) as well as mostly symmetrical entities (marmalade box, mug, flower cup, pitcher) and more complex examples (spray flask, toy car). This allows to generalize and evaluate the amount of information that can be obtained from a single tactile exploration for different objects.

After some experimentation with different fitness functions the form shown in (1) was found best. However, even this function suffers from local minima and after initial experiments, the Steepest Descent method was found to be unable to converge and thus unsuitable and was dropped from the full experimentation. For experiments with the full set of objects, it was decided to use the three metaheuristics: SA, PSO and FF algorithms. This allows us to ensure that results are not dependent on the performance of a particular method.

B. Simulator

Transportation of an object is chosen as a practical task for the robot in this study. The scenario is that a robot should grasp an object and move it to another location on a table. OpenRAVE simulator [28] was chosen as the simulation environment. This simulator provides a wide range of grasp and manipulation capabilities, including different grasp quality metrics and path planning algorithms. The optimization algorithms were implemented in MATLAB, which integrates to OpenRAVE through scripting.

Barrett WAM arm with the Barrett hand depicted in Fig.3 was used in all experiments. It is one of the most popular 3-fingered robots used in grasping and manipulation research. WAM arm has 7 DOF and Barrett hand has 4 DOF (spread

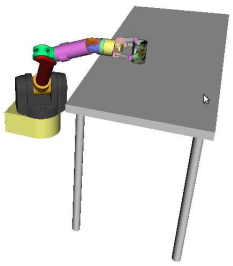


Fig. 3: Barrett WAM robot

angle of the fingers and 3 angles of proximal links). The contact information was detected at the colored parts of the hand shown in Fig.4.

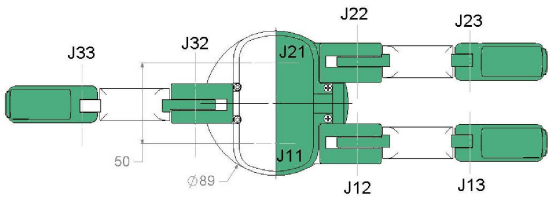


Fig. 4: Barrett hand with contact areas

C. Results

We carried out Experiment 1 performing an optimization after a single grasping attempt for 3DOF case having uncertainty in location and orientation parameters using the three metaheuristics described in Section IV. The search region was limited by $[0.1 \quad 0.3]$ meters in x direction, $[-0.2 \quad 0.0]$ meters in y direction. The range of variations in angle of rotation around z-axis was set to 90 degrees. From each simulation run a binary collision matrix was obtained. Each value of this matrix corresponds to existence or absence of a contact between one of the robot hand links and a graspable object at a defined time instant. Results for the experiment

Preprint submitted to Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics. Received January 27, 2012.

TABLE I: Results for Experiment 1

Object	Error		
	coordinate	angle	fitness
standard box	0.00411	5.22	9.00E-005
tea box	0.01936	4.38	7.80E-004
marmalade box	0.00117	26.70	6.00E-005
standard mug	0.00040	1.80	3.50E-005
flower cup	0.00122	1.13	3.00E-005
pitcher	0.00051	14.25	3.00E-005
spray flask	0.00827	11.91	3.00E-005
toy car	0.00313	4.59	3.00E-005

TABLE II: Experiment 2 results

Object	Error types	Results
standard box	coordinate	0.00161
	angle	0.04
	scale	0.0009
	fitness	3.00E-005
flower cup	coordinate	0.00298
	angle	1.11
	scale	0.0437
	fitness	5.90E-003

using PSO approach are presented in Table I. The table shows errors for parameters and fitness function values for eight test objects. The errors indicate differences between actual object's attributes and the results of optimization.

The table contains only results for PSO approach, because this method showed consistently good results for all objects. For most of the objects, the final fitness function value was the smallest among the three optimization approaches. SA algorithm also provided reasonably precise results and its run-time was smaller than for PSO. FF method, in general, produced less accurate results. As already noted, the gradient-based approach did not manage to converge in most cases. This can be explained by the fitness function shape, shown for a 2-D case in Fig.5. As can be seen from the plot there are several local minima and a large flat area, which the Steepest Descent approach is not able to overcome.

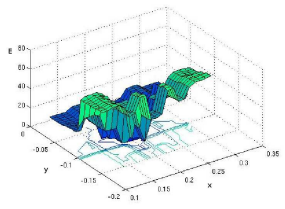


Fig. 5: Fitness function shape

For the positional parameters, the errors in x and y coordinates were within a couple of millimeters for all tested objects for all metaheuristics approaches. In contrast, the experiments reveal that the angular error is highly dependent on the object geometry. For mostly symmetrical objects such as the marmalade box it is impossible to determine an exact angle. Even for complex-shaped bodies such as the spray flask and toy car an orientation error is relatively large. Such results can be explained by the fact that there is a complex interplay between the Cartesian and angular position variables affecting the fitness function value. If the Cartesian position is known, we can often obtain quite precise angular information. However, in most of cases such information is not available. To obtain better orientation results, the action would need to be planned in order to maximize the information gain. This stage would be especially important for objects which have asymmetrical features only on small surface patches (cup, mug, pitcher). The exploration should be planned such that these asymmetries (tip of pitcher, handles of pitcher and cup) of the graspable objects would be detected by tactile sensors. However, those exploration attempts would seldom result in stable grasps, and would then require actions dedicated to exploration, in contrast to our approach, where the exploration is a by-product of a grasp attempt. Moreover, planning of exploration in uncertain conditions is a non-trivial task, which would likely require a construction of probabilistic models. It was thus left out of the scope of this study.

It is important to note that imprecise angle determination may not be a serious problem in some applications. For example, grasp stability and transportation task execution for symmetrical objects are not affected by the angle of rotation. For simple manipulation and transportation tasks it is usually sufficient to grasp the body from above without taking into account its orientation. On the other hand, if the task, for example, is to pour water from the pitcher, the determination of the handle location is a crucial objective.

Regardless, while the angular error values for majority of objects are somewhat significant, the fitness function values are small. Thus, it can be noted that values of fitness function are mostly dependent on location rather than orientation. The possibility to learn about the object orientation from a single grasp, thereby, is highly object and hand dependent. There are examples when problems can occur not only with angle determination, but also in estimation of other parameters. For instance, in the case of grasping an elongated object in the middle, there is a manifold of equally good solutions along the object. All grasping attempts along this direction will result in equal contact information. Thus, it can be concluded, that the tactile readings received during a single manipulation action carry quite limited amount of information about an object.

In Experiment 2, the approach was tested in 4DOF case (uncertainty in x,y, angle of rotation and scale factor) on 2 objects (standard box and flower cup). Results for the experiment with SA algorithm are given in Table II. SA method was chosen as in 3DOF case it was found to provide good results with smaller computational effort compared to PSO. The results show that the optimization approach works well, the size information can be estimated very precisely and the fitness function error in this case is small. Not only coordinate and scale, but also angle of rotation values are quite close to real.

Preprint submitted to Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics. Received January 27, 2012.

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

We have presented a novel approach using simulation to answer the question: how much information about the envicontrol of manipulationronment can be obtained from tactile sensors during a single manipulation attempt? The developed technique is based on minimizing the difference between predicted and real sensor measurements. A series of experiments was conducted in simulation using the 3-fingered Barrett WAM robot model. A transportation task was considered for several objects of various shapes for three types of unknown attributes: location, orientation and size. To increase the generality of the results, three different optimization metaheuristics were used.

Our main conclusion is that learning about the environment using the tactile sense during manipulation is surprisingly difficult, even in optimal conditions without any sensor noise. Moreover, the difficulty of estimating different attributes varies significantly: experimental results showed that object location and scale factor can often be estimated relatively well, but that the accuracy of orientation estimation is very object dependent. More generally, the entire estimation problem can be ambiguous, for example, due to symmetries in object shape.

B. Future Work

To achieve accurate estimates for all parameters, it would be possible to plan initially for object exploration instead than for the manipulation action. However, planning for the exploration under uncertainty requires further work. A related approach would be to integrate multiple tactile exploration actions. A different direction of research would be to modify the hardware and sensors. For example, use of dexterous hands with a large number of fingers could improve results due to the possibility to collect more contact information from a single manipulation attempt. Finally, it would be interesting to explore the viability of the optimization approach in a setting with a real robot, even if this is out of the scope for the current paper, where we are interested in finding the limitations of the sensing in optimal conditions.

REFERENCES

[1] J.R. Flanagan, M.C. Bowman, and R.S. Johansson. *Control strategies in object manipulation tasks.* Current Opinion in Neurobiology, 2006, pp. 650-659

[2] W.D. Hillis, "Active touch sensing", *Int J. Robot. Res.*, vol.1, no.2, 1982, pp. 33–34.

[3] R.A. Russel, "Object recognition by a 'smart' tactile sensor". In *Proceedings of the Australian Conference on Robotics*, 1984, pp. 93–98.

[4] N. Gorges, S.E. Navarro, D. Göger and H. Wörn. Haptic Object Recognition Using Passive Joints and Haptic Key Features. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[5] K. Hsiao, L. Kaelbling and T. Lozano-Perez. "Task-Driven Tactile Exploration". In *Proceedings of Robotics: Science and Systems*, 2010.

[6] H. Dang, J. Weisz and P.K. Allen, *Blind Grasping: Stable Robotic Grasping Using Tactile Feedback and Hand Kinematics*, ICRA 2011, Shanghai, 2011.

[7] R.D. Howe, "Tactile sensing and control of robotics manipulation", *J. Adv. Robot.*, vol.5, 1994, pp. 245–261.

[8] Z. Doulgeri and Y. Karayiannidis, "Force position control for a robot finger with a soft tip and kinematic uncertainties", *Robot. Auton. Syst.*, vol.55, no. 4, 2007, pp. 328–336.

[9] P.A. Schmidt, E. Maël and R. P. Würtz, "A sensor for dynamic tactile information with applications in human-robot interaction and object exploration", *Robotics and Autonomous Systems*, vol. 54, no. 12, 2006, pp. 1005–1014.

[10] S. Caselli, C. Magnanini and F. Zanichelli, Haptic Object Recognition with a Dextrous Hand Based on Volumetric Shape Representations .In *Proc. of the IEEE Int. Conf. on Multisensor Fusion and Integration, Las Vegas, Nev.*, 1994, pp. 2-5.

[11] N. Gorges, S. E. Navarro and D. Göger and H. Wörn, "Haptic Object Recognition Using Passive Joints and Haptic Key Features". In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[12] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt and W. Burgard, "Object Identification with Tactile Sensors Using Bag-of-Features". In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2009.

[13] Z. Pezzementi, E. Plaku, C. Reyda and G.D. Hager, "Tactile-Object Recognition From Appearance Information", *Robotics, IEEE Transactions*, vol. 27, no. 3, 2011, pp. 473–487.

[14] A. Petrovskaya and O. Khatib, "Global Localization of Objects via Touch", *Robotics, IEEE Transactions on*, vol.27, no. 3, 2011, pp. 569–585.

[15] S. Chitta, M. Piccoli and J. Sturm, "Tactile Object Class and Internal State Recognition for Mobile Manipulation". In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, pp. 2342–2348.

[16] Y. Bekiroglu, J. Laaksonen, J.A. Jorgensen, V. Kyrki and D. Kragic, "Assessing Grasp Stability Based on Learning and Haptic Data", *Robotics, IEEE Transactions on*, vol. 27, no. 3, 2011, pp.616–629.

[17] K.T. Gunnarsson and F.B. Prinz, "CAD Model-Based Localization of Parts in Manufacturing", *Computer*, vol. 20, no. 8 1987, pp. 66–74.

[18] K.T. Gunnarsson, "Optimal part localization by data base matching with sparse and dense data", Ph.D. dissertation, Dept. of Mechanical Engineering, Carnegie Mellon Univ., 1987.

[19] Y. X. Chu, "Workpiece Localization: Theory, Algorithms and Implementation", Ph.D. thesis, HKUST, 1999.

[20] A. Petrovskaya, O. Khatib, S. Thrun and A.Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors". In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference*, 2006, pp. 707–714.

[21] J. Laaksonen and V. Kyrki, *Probabilistic Approach to Sensor-based Grasping*, ICRA 2011 Workshop: Manipulation Under Uncertainty, 2011.

[22] A.T. Miller, *Graspit!: A versatile simulator for robotic grasping*, IEEE Robotics and Automation Magazine, 11:110122, 2004.

[23] W.H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in Fortran 77*, Cambridge University Press, second edition, 1992.

[24] J. Kennedy and R. C. Eberhart, Particle swarm optimization. In *Proc. of IEEE International Conference on Neural Networks*, Piscataway, NJ.,1995, pp. 1942–1948.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*, Science, 1983, pp. 671–680.

[26] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, "Equations of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, 1953, pp. 108–1092.

[27] X. S. Yang. *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK, 2008, pp. 79–90.

[28] R. Diankov, J. Kuffner, "Openrave: A planning architecture for autonomous robotics", Technical Report CMU-RI-TR-08-34, Robotics Institute, Pittsburgh, PA, July 2008.

[29] L. Zhu, H. Luo and H.Ding "Optimal design of measurement point layout for workpiece localization",*Journal of Manufacturing Science and Engineering 131*, 2009.

[30] K. C. Sahoo and C.H. Menq "Localization of 3-D objects having complex sculptured surfaces using tactile sensing and surface description", *ASME Journal of engineering for industry*, 113(1), 1991, pp. 85–92.

Preprint submitted to Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics. Received January 27, 2012.

Probabilistic Sensor-based Grasping

Janne Laaksonen and Ville Kyrki

Abstract— **In this paper, we present a novel probabilistic framework for grasping. In the framework, grasp and object attributes, on-line sensor information and the stability of a grasp are all considered through probabilistic models. We describe how sensor-based grasp planning can be formulated in a probabilistic framework and how information about object attributes can be updated simultaneously using on-line sensor information gained during grasping. The framework is demonstrated by building the necessary probabilistic models using Gaussian Process regression, and using the models with an MCMC approach to estimate a target object's pose and grasp stability during grasp attempts. The framework is also demonstrated on a real robotic platform.**

I. INTRODUCTION

Current grasp planning approaches are usually based on an assumption of perfect knowledge of target objects. While geometric models are good approximations of the objects in the real world, the models are not exactly accurate, especially when speaking of household items. Thus, a difference between the expected and the realized grasp arises from these approximations, although in many cases the difference is small enough to achieve a stable grasp. However, this discrepancy is usually left unused.

On the other hand, methods utilizing sensor information to grasp using corrective motions or reacting to the tactile sensor information have been proposed. Contrary to grasp planners, accurate object models are not usually available in this type of grasping. However, it has been shown that using for example tactile sensors it is possible to estimate stability of the grasp [1], the pose of the object [2], [3], or even the identity of the grasped object [4].

In this paper, we present a probabilistic framework, which unifies the ideas behind grasp planning and the possibilities of sensor-based grasping. The framework considers the required variables and models for grasping as probability distributions and allows thus the representation of the current belief probabilistically, that is, the uncertainty in the knowledge can be represented. The framework allows interplay between grasp planning and corrective motions, in situations where object attributes, such as pose, are not precisely known, by utilizing sensor information gained during grasping. Such a situation can arise for example when visual sensing is used to initially estimate the target objects.

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821.

J. Laaksonen and V. Kyrki are with Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland, jalaakso@lut.fi, kyrki@lut.fi

We experimentally demonstrate and study the framework using simulations and on an actual robot. In the demonstration, we use Metropolis algorithm, a MCMC (Monte Carlo Markov Chain) method, to model the evolving probability distributions. A Bayesian approach is used, that is, instead of using the maximum likelihood or maximum a posteriori solution, the result is obtained by marginalizing over the current knowledge. The experiments show that the proposed approach can perform grasp planning with uncertainties in environment as well as measurements, for example, if the target object's pose is uncertain.

Section II collects the related work about grasp planning and other related fields and Section III describes the probabilistic framework. In Section IV, we present an approach on how to model key object properties using Gaussian processes. In Section V, a practical implementation, based on the probabilistic framework, is presented where we simulate a Barrett hand grasping several objects. We also demonstrate the framework with a real platform. We conclude with discussion and possibilities of the probabilistic framework, in addition to our focus of future work in Section VI.

II. RELATED WORK

Our approach to find good grasps is closely related to the field of grasp planning. In grasp planning the goal is to find as good as possible grasp on a given object. The goodness of the grasp is usually measured with a grasp quality measure [5]. However, compared to our method, most current grasp planning methods do not account for the uncertainty present in the object's shape or pose information. Also most of the grasp planning methods require a known geometric model of the object.

To simplify the grasp planning, many methods employ some form of object decomposition. The goal of the decomposition is to reduce the amount of feasible grasps without trying every grasp on an object. In [6], the object is decomposed to minimum volume bounding boxes, in an effort to understand the underlying shape of the object. The primitive shape is then used to reduce the search space for stable grasps. Instead of boxes, superquadrics are used in [7]. In addition to the construction of the superquadric decomposition, heuristic is used to define the trial grasps based on the superquadric form of the object, limiting the space of grasps significantly.

The Columbia Grasp Database [8] takes a different approach to most grasp planners and computes best grasps for a set of hundreds of objects. The grasp planning problem is then transformed to a problem of matching a new object with

an object found in the precomputed database of grasps. The work has also been extended to consider partial data [9].

If the object is not known, i.e. a geometric model is not available, the grasp planning methods can still be used if the model of the object can be constructed. The model construction can either be done by vision or tactile exploration. However, the geometric model in this case is usually a mesh or a point cloud, and contains no information about the inherent uncertainty related to the perception. Approaches such as [6] can be applied here as well but the results can be worse than in the cases where the full geometric model is known. Moreover, the decomposition may fail in cases where large volumes are missing from the perceived object.

Another approach for finding grasps is object affordance modeling. While object affordance is a broader subject, the affordances can also be thought in the sense of grasp stability. In some of the grasp related studies, grasp affordances consider the overall stability of the grasp [10], [11] or, for example, the grasp affordance in specific tasks [12].

Learning to find good grasps is another view on the problem. [10] utilizes learning on a real robot to learn the grasp affordances of an object. The learning process reduces a vision bootstrapped distribution of grasps to a smaller set of grasps containing only good grasps. Reinforcement learning [13] can also be applied, so that a sequence of grasps can be learned which will lead to a stable grasp of an object.

Our approach to grasping is more related to the methods found in [14], [2]. The aim of [14] is to reduce the uncertainty of an object's pose to enable grasping the object. In [2], the shape of the object is also uncertain in addition to the pose. In both studies, the method is presented with a parallel jaw gripper grasping a 2D-object. However, these methods do not utilize sensor information gained during grasping. Also in [15], the authors propose a decision-theoretic controller which minimizes the uncertainty of the object pose using arm trajectories to enable task specific grasps on objects. Tactile sensors were used to detect contacts between the hand and the objects. A new algorithm, Guaranteed Recursive Adaptive Bounding (GRAB), for inference was developed in [16]. The algorithm was also tested in a manipulation environment where the method made accurate inference of object's pose in both simulation and real environments. The method was further developed in [17]. However, only the problem of object localization was studied. Key difference of our method with both [15] and [17] is that we estimate the object pose without a geometric model and neither methods use grasp stability to direct the grasping actions.

This paper presents a novel probabilistic framework for reasoning about the grasp stability and the object attributes, so that grasp planning can be performed even if object attributes are uncertain. Initial work about the framework has been published in [18], which the current paper substantially develops by introducing probabilistic object modeling instead of using simulation, by extending the demonstration to 3D-objects, and by demonstrating the framework with a real robotic platform.

III. GRASPING IN A PROBABILISTIC FRAMEWORK

The probabilistic framework is now presented in a general form. We model sensor-based grasping using the following variables: S denotes the stability of a grasp as a binary value, G the grasp attributes (e.g. the pose of the end-effector), O the object attributes (e.g. the pose of the target object) and T represents on-line measurements, for example, tactile information. The variables have characteristics: G, the grasp attributes, can be controlled, T can be measured for each grasp attempt, while O is uncertain, that is, we assume we only have an uncertain initial estimate of the object attributes.

Traditional grasp planning algorithms can be interpreted in a probabilistic framework as attempting to maximize the stability, S, by controlling the grasp attributes, G, with perfect knowledge of the object attributes O,

$$\max_{G} \; P(S|G,O) \; . \tag{1}$$

In our model, O is not assumed to be precisely known but instead it is represented as a probability distribution. Moreover, we do not assume that the available model (1) is precise, that is, the stability of a grasp given information about the object and grasp need not be exact, but instead the model itself can exhibit uncertainty, for example, due to simplifications made in a simulator to compute a grasp quality metric.

It has been shown that grasp stability can be estimated using tactile information [1]. Thus, we can build a probabilistic model for the stability given the other variables, $P(S|G,O,T)$. That model can be used to assess the stability of a single grasp attempt, as shown in [1]. Moreover, for stability detection with uncertain object knowledge, we can marginalize over the uncertain object attributes, such that the probability of a stable grasp given the grasp attributes and tactile measurements is given by

$$P(S|G,T) = \int P(S|G,O,T)P(O|G,T)\,\mathrm{d}O \; . \tag{2}$$

If the grasp attributes are also uncertain, we can marginalize over them in a similar fashion to find $P(S|T)$. This is also the model for grasp stability for the case where no information about the object or grasp is used for stability recognition.

In order to perform grasp planning, we again need to marginalize over the distribution of object attributes. That is we need to find the mode of $P(S|G)$. The marginalization can be written

$$P(S|G) = \int P(S|G,O)P(O)\,\mathrm{d}O \; . \tag{3}$$

This is a major difference to traditional grasp planning, where the best single estimate of object attributes, that is, the mode of O, is used instead of marginalization over the whole distribution.

Because the tactile information for a grasp attempt is not available before the attempt is performed, we can use the tactile information from the previous grasp attempts only to update the posterior distribution for the object attributes

$P(O)$. That is, we can use the model $P(O|G,T)$ to update the posterior of object attributes. Thus, after some tactile information has been collected, for grasp planning we find the maximum

$$\begin{aligned} &\arg\max_G P(S|G) \approx \\ &\arg\max_G \int P(S|G,O)P(O|G_{t_0:t-1},T_{t_0:t-1})\,\mathrm{d}O\ . \end{aligned} \tag{4}$$

This shows that the stability S can be maximized by finding the best grasp G, when $G_{t_0:t-1}$ and $T_{t_0:t-1}$ are known (subscripts denoting that these are from the previous attempts).

The process described by (4) is depicted in Figure 1. From the figure it can be seen that the knowledge of the object attributes O, is iterated over the time steps, $t_0, \ldots, t-1, t, t+1, \ldots, t_n$. The knowledge of O is refined using information from the known grasp attributes G and the measurements T.

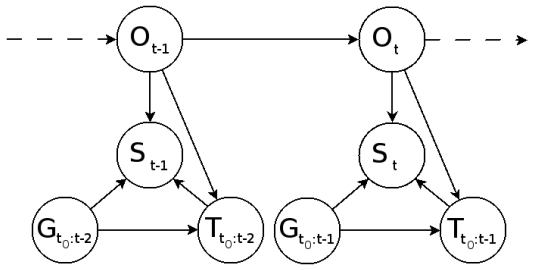


Fig. 1. Process of refining object knowledge.

To build a working system based on the Equation (4), two models are needed:

- Model for $P(O|G,T)$, describing relation between tactile information and grasp and object attributes.
- Model for $P(S|G,O)$, stability as a function of grasp and object attributes

These models are not trivial to build and depend on the object and the manipulator used to grasp the object. Still, there exists models for both cases, e.g. see [3] for a model for $P(O|G,T)$ and [1] for a model for $P(S|G,O,T)$. It should also be noted that $P(O|G,T)$ can be obtained from a prediction model of sensor measurements $P(T|G,O)$ using the Bayes formula. One approach to generate the models is to simulate the object and the manipulator to produce the required tactile information and stability models. We have used this approach to demonstrate the framework in Section V.

Our framework does not place constraints on the actual models, and the attributes G, O, T can be freely chosen. For example, G and O can include the poses of the manipulator and the object. The benefit of the presented probabilistic framework is that throughout the grasping process the uncertainty of the actions arising from equation (4) is known. Also, measurement errors can be accounted for during both grasp planning as well as on-line grasp stability detection.

IV. Modeling

As mentioned in Section III, two models are required to successfully utilize the framework described in this paper: one to model the object attributes given grasp attributes and tactile information and another model for grasp stability given object and grasp attributes.

In simulation, the modeling is relatively easy as we have perfect knowledge of the object and the manipulator grasping the object. However, our goal is to have models that can be used outside of the simulator. Another practical requirement for the models is that the models must be generative to account for the whole state space. For these reasons, we build the required models using Gaussian Process Regression (GPR).

The Gaussian Process (GP) used in GPR is defined as a set of random variables of which any finite number have a joint Gaussian distribution. The GP is constructed of the mean function $m(x)$ and the covariance function $k(x,x')$. As GP models the data using these functions, the GP is able to model the underlying uncertainty present in the data, which grows in the gaps of the data and lessens where the data has high density.

Suitable mean and covariance functions are usually dependent on the form of the data and requires some thought. Different covariance functions enable different type of data to be modelled. In the case of grasping, many discontinuities appear due to discrete events such as a finger missing an object completely during grasping. Due to this we have chosen to use the neural network covariance function, which is non-stationary, and can model discontinuous data better than the more commonly used squared exponential covariance function [19]. Choosing which mean function to use is not as critical to the regression as the covariance function.

Once the model has been selected, GPR finds the most probable function over the training data. GPR can then be used to estimate $f(x)$ given new x. However, to operate properly GP requires that hyperparameters, i.e. parameters of the mean and covariance functions, are set as well. Finding proper values for the hyperparameters is the most challenging task with GP techniques but methods exist that use training data to optimize the hyperparameters. Readers interested in GPs can get a deeper understanding from e.g. [19].

V. Experiments

To demonstrate the viability of the framework presented in Section III, we have estimated the models $P(O|G,T)$ and $P(S|G,O)$ using data from simulations. Our goal is to show that we can improve the estimate of uncertain object attributes while simultaneously improving the grasp stability given our estimate of the object's attributes using sparse tactile information received during grasp attempts. As the tactile measurements, T, we use only the joint configuration of the robot hand. We use both simulation and real robot to validate our approach. The simulation setup is described in Sections V-A and Section V-B and the setup for the real robot is depicted in Section V-D.2.

A. Experimental Setup

We use GraspIt!-simulator as our simulation environment [20]. Figure 3 shows the Barrett hand and the four

objects used in the experiments. All of the objects fit inside the grasp of the Barrett hand. The objects are assumed to lie on a planar surface, thus having two-dimensional spatial uncertainty represented by three variables forming a tuple (x, z, θ), where x and z are Cartesian coordinates in millimeters and θ is the orientation in degrees. We employ top grasps and thus the hand is also moved only along the same three dimensions. However, both the hand and the objects have full 3-D geometric models. Furthermore, we assume that the object we grasp remains stationary throughout the grasps. While this may seem contrary to experiences with real robotic hands, we argue that with good sensors, such as [21], the grasping can be controlled so that sufficiently massive objects do not move significantly during grasping. The framework can accommodate non-stationary objects, by using only the most recent measurement of T, as in [18], or by using a motion model for $P(O_{t+1}|O)$. However in this paper we focus on stationary objects.

B. Data Collection and Modeling

GPR, introduced in Section IV, is used to model both $P(O|G,T)$ and $P(S|G,O)$. The goal of the approach is to model the joint configuration $T|G,O$, and the quality of the grasp $S|G,O$, when given the relative pose between the object and the hand. For the quality measure we utilized the existing quality measures in GraspIt! and chose the force-closure quality measure, i.e. the ϵ-measure. In this paper, we consider quality measure value of 0.1 or greater to be stable.

The data used to optimize the GPR was generated using a grid in (x, z, θ) and then the object was grasped at each of the points in the grid. For the cylinder and cube, x and z were discretized from -50 mm to 50 mm at 10 mm intervals and θ was discretized from -180 to 180 degrees at 15 degree interval. For the two mugs, x and z were discretized from -100 mm to 100 mm at 20 mm intervals and θ was discretized from -180 to 180 degrees at 15 degree interval. The grid was centered on the object. Each grasp was executed using the auto grasp-function present in GraspIt!, thus, only one preshape was used when grasping. After each grasp the quality of the grasp was measured and the finger joint values were recorded with the relative pose. We have also restricted the search space to these boundaries in the implementation.

C. Implementation

Our general approach is based on the sequence of actions shown in Figure 2. We assume that some type of initial estimate (with associated uncertainty) of the object pose is obtained in phase 1, e.g., from vision. Using the estimate, we can plan for a grasp with the uncertainty from the initial estimate, phase 2. Then a grasp is performed, phase 3, giving measurement data (we assume that at least joint configuration data is available). Using the measurement data, we can make a decision of the grasp stability, phase 4. If the grasp is stable, the object can be manipulated, if not, we can plan for a new grasp, phase 5, with the new information from the attempted grasp. This loop can then be further iterated until grasp stability conditions are satisfied.

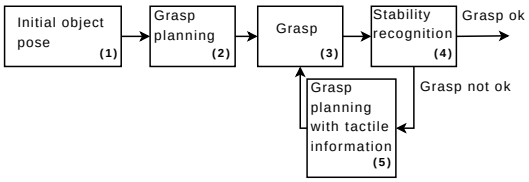


Fig. 2. Sequence of actions.

The theoretical framework described in Section III is implemented with MCMC methods. The object attribute probability distribution, $P(O|G,T)$, is modelled with Metropolis algorithm while a particle filter-based maximization is used to search the maximum of the probability distribution of stability, $P(S|G,O)$. Both methods model the probability distributions with a cloud of particles to make the computation of evolving probability distributions tractable. We have chosen the Metropolis algorithm for the object attribute probability distribution because the method allows modeling of the whole distribution without the degeneracy problems occuring with particle filters. More information on particle filtering, especially applied to robotics can be found in [22]. Particle filters have been used in manipulation, for example in [3], to estimate object pose using tactile sensors.

Algorithms 1 and 2 describe our method of finding stable grasps. Algorithm 1 requires the initial estimates of the uncertainty, given in $\sigma_{\mathbf{init}}$, for each of the variables (x, z, θ). The particle set O_1 in Algorithm 1 represents the probability distribution of the object, $P(O|G,T)$. Note that in line 10, we weigh the particles in O_1 with all the previous grasps. Also in line 10, the likelihood $p(J_k|J_k^*)$ is computed given the actual joint configuration J_k^* and the joint configuration J_k given by GPR. The likelihood function is simply a Gaussian function centered at J_k^*, with a variance obtained from GPR for each individual sample. The probability for a stable grasp, $\sum_i P(S|G,O_{1_i})$, determines how well the object must be localized, and can be used to force the system to make more grasps until the uncertainty of the object's pose is small enough to attain the desired probability, $\text{thr}_{\text{stable}}$, for a stable grasp.

Particle set G_1 in Algorithm 2 represents all the possible grasps, and by applying the grasp motion induced by each grasp particle to each of the particles in O_1, we can find the probability of a stable grasp $P(S|G,O)$. Quality of the grasp for each individual grasp and object pose combination is queried from the GPR model $S|G,O$. In Algorithm 2 the maximum of distribution, $\arg\max_G \int P(S|G,O)$, is searched for and the corresponding motion is then applied.

Relating to Figure 2, Algorithm 2 takes care of the grasp planning, that is, phases 2 and 5. Algorithm 1 handles phase 3, grasping the object and updating the belief of object pose. In line 13 of Algorithm 1, the grasp stability probability is computed and corresponds to phase 4 of the action sequence.

Algorithm 1 find_stable_grasp($\mu_{\mathbf{init}}$,$\sigma_{\mathbf{init}}$)

Generate initial particle set, $O_1 \sim \mathcal{N}(\mu_{\mathbf{init}}, {\sigma_{\mathbf{init}}}^2)$
$t \leftarrow 0$
while Grasp not stable **do**
  $(x,\ z,\ \theta) \leftarrow$ **find_best_grasp**(O_1,σ_{init})
  Move hand to $(x,\ z,\ \theta)$
  Grasp object, store joint configuration as J_t^*
  while O_1 is not converged **do**
    For each particle i in set O_1, generate a proposal i_p using distribution $\mathcal{N}(0, {\sigma_{\mathbf{1}}}^2)$ with $\sigma_{\mathbf{1}} \leftarrow (\sigma_x,\ \sigma_z,\ \sigma_\theta)$
    For all i and i_p, query GPR model $T|G, O_1$ for joint configurations $J_{0,\cdots,t}$
    For all i and i_p, compute posterior probability $\propto \prod_{k=0}^{t} p(J_k|J_k^*)$
    Choose according to the Metropolis algorithm whether to replace i with i_p
  end while
  Approximate $P(S|G,O)$ by $\sum_i P(S|G,O_{1_i})$
  if $\sum_i P(S|G,O_{1_i}) > \text{thr}_{\text{stable}}$ **then**
    Grasp is stable
  end if
  $t \leftarrow t+1$
end while

Algorithm 2 find_best_grasp(O_1,σ_{init})

Generate particle set, $G_1 \sim \mathcal{N}(\mu_{O_1}, \mathbf{5}{\sigma_{\mathbf{O_1}}}^2)$, where μ_{O_1} is the mean of O_1 and σ_{O_1} is the standard deviation of O_1
while G_1 is not converged **do**
  Weigh particles G_1, $w_2 \propto P(S|G,O_1)$
  $(x_{max}, z_{max}, \theta_{max}) \leftarrow \arg\max_G P(S|G,O_1)$
  Do importance filtering according to w_2
  Use $\mathcal{N}(0, {\sigma_{\mathbf{2}}}^2)$ as proposal distribution with $\sigma_{\mathbf{2}} \leftarrow 0.2\ \sigma_{\mathbf{init}}$
end while
return $(x_{max}, z_{max}, \theta_{max})$

D. Experiments

In the experiments, we show a sequence of grasps, following the diagram in Figure 2. We demonstrate that we can localize objects using the Metropolis algorithm despite uncertain initial estimate during several grasp attemps. In the experiments, 1000 particles were used to model the object attribute distribution, and 100 particles to find the maximum of the grasp stability.

1) Grasping an object with uncertain pose: In this experiment we show for two primitive objects, the cube and the cylinder, and two complex objects, mug 1 and mug 2, shown in Figure 3, that we can obtain a stable grasp even if the initial estimate of the object's pose is uncertain. Each grasp also improves the localization of the object.

In the experiments we assume the object mean $\mu_o = (0,0,0)$ and object standard deviation $\sigma_o = (40,40,40)$. The chosen σ_o shows that we are highly uncertain of the initial pose of the object. μ_o and σ_o are given as input to Algorithm 1. In the experiments we also limited the number of grasps to 4, instead of defining a threshold for the stable grasp probability. In Algorithm 1, the object posterior distribution refinement was limited to 50 iterations instead of a more sophisticated convergence criterion. However, 50 iterations were found to form the posterior distribution adequately.

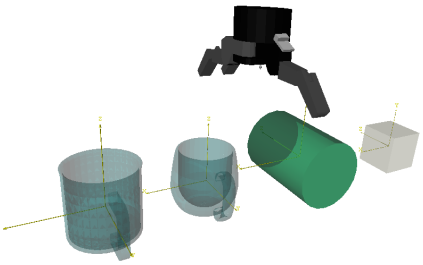


Fig. 3. Objects used in the experiments, from the left: mug 2, mug 1, cylinder and cube. The simulated Barrett-hand is also shown in the figure.

Due to the probabilistic nature of the algorithms we show the results as examples of posterior distributions, which can be seen in Figure 4 for two different runs of Algorithm 1. Each run in Figure 4 shows how the posterior distribution of the object pose evolves after grasp attempts are made, each run consists of 4 grasps. The real object pose is marked with a red cross in each subfigure. Title of each subfigure in the figure shows the quality measure (QM) computed by GraspIt! and the corresponding stable grasp probability (Stab. Prob). Quality measure of -1 indicates a non-force-closure grasp.

First run, in Figure 4(a)-(d), shows the result for the cube visible in Figure 3. The distributions show that multiple modes are found near the correct pose. The probability of a stable grasp increases after each successive grasp, due to the convergence of the object pose posterior. The quality measure also shows that a force-closure is achieved during the grasps. In the second run, in Figure 4(e)-(h), mug 2 is grasped. The posterior distribution reflects the uncertainty in orientation of the mug, as the mug is almost completely symmetric. As before, the grasp stability probability increases after each grasp. The quality measure is also improved. However, the posterior density is not as dense as in Figure 4(a)-(d).

All of the posterior distributions show outliers at the edges of the search space. This is due to the imperfect regression results by GPR. One of the probable reasons is the discretization of the sampling grid which is quite coarse.

To show that the probabilistic grasp planning shown in Algorithms 1 and 2 actually improves the grasp quality, we ran multiple tests on all 4 objects to find out how each successive grasp reduces unstable grasps by refining the knowledge of the object pose. Results are described in Table I. The table shows the probability of achieving a stable grasp after the number of grasps shown in the "Grasp"

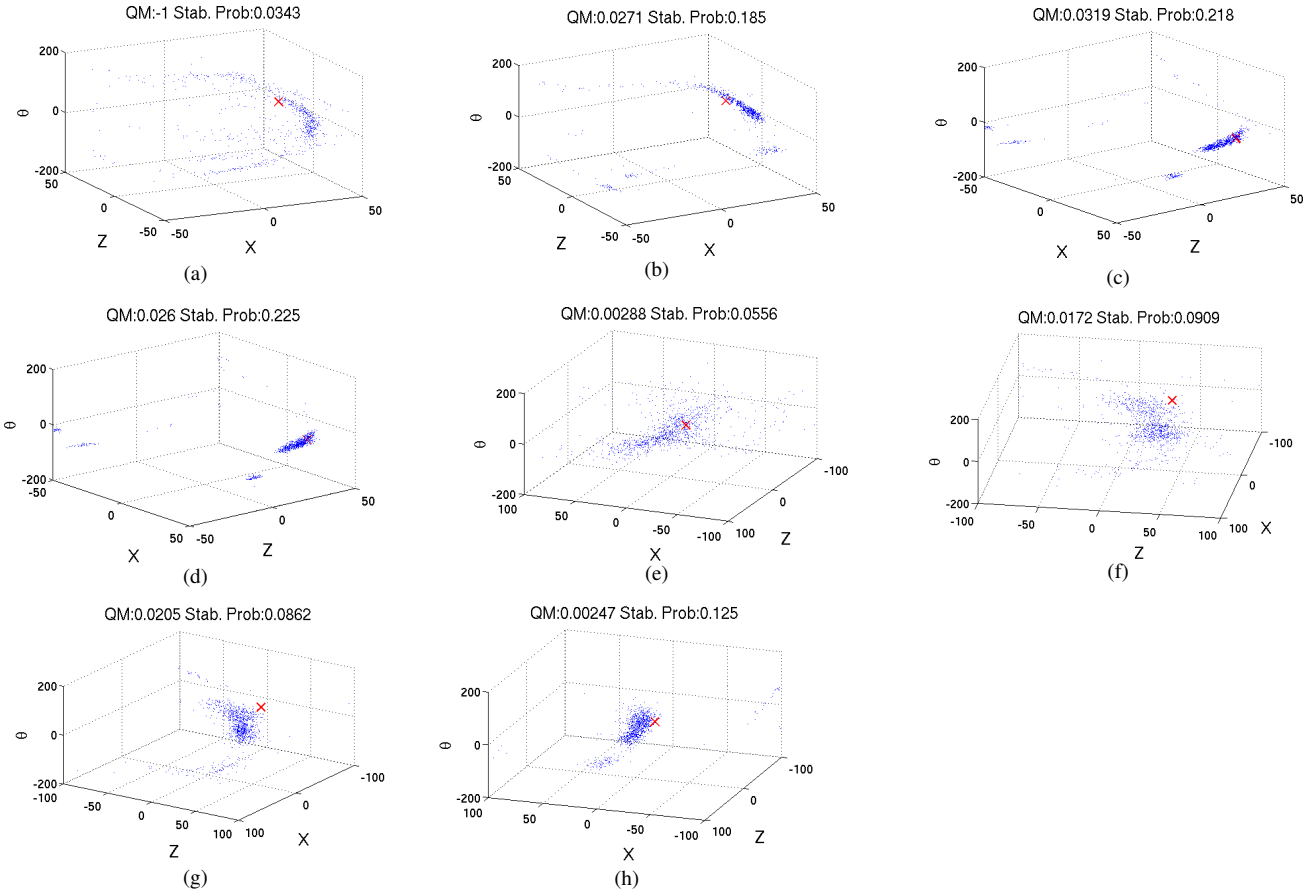


Fig. 4. (a)-(d): Grasping the cube at pose (45,25,0);(e)-(h): Grasping the mug 2 at pose (-32,38,68).

TABLE I

PROBABILITY OF A FORCE-CLOSURE GRASP ACROSS 30 RUNS.

Grasp	Cube	Cyl.	Mug1	Mug2
1	0.10	0.03	0.07	0.43
2	0.90	0.63	0.80	0.63
3	0.93	0.23	0.77	0.83
4	0.93	0.70	0.67	0.80

column. From the results, one can see that the framework is able to refine the object pose distribution and as a result find a stable grasp more often. The results also show that typically the probability of stable grasp occurring is increased significantly after the first grasp. This result shows that it is not necessary to have the object stationary during grasping, as the second grasp is probable to achieve stable grasps in most cases.

2) Experiments with a real platform: To validate the approach proposed in this paper, the framework was implemented on a robotic platform, consisting of Melfa RV-3SB 6-DOF arm and Weiss WRT-102 robotic gripper with integrated tactile sensors. Using the approach presented in V-B, we sampled the object shown in Figure 5 with the gripper. To keep the object stationary the object was suspended on three screws glued to the object. Due to the limitations of our gripper we decreased the DOFs from three to two, (x, θ). The dimensions of the object were 45 mm across Y and 120 mm across X. We collected 26 samples from the object,by grasping from -50 mm to 90 mm in X and from -7 degrees to 7 degrees in θ. Additionally, we utilized reactive grasping to enable grasping of the object, as we can only control the width of the grip, which was our measurement T. The stability was determined manually by lifting the object after each grasp and labeling each grasp as stable or unstable.

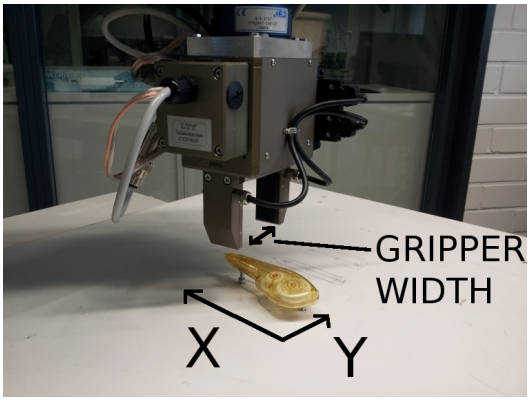
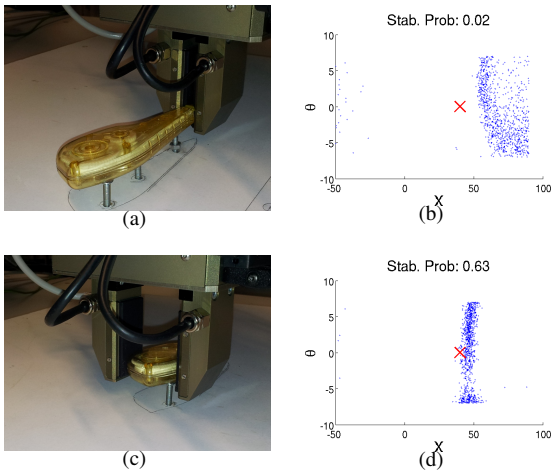


Fig. 5. The robot gripper and the object, a correction roller.



(a) (b) (c) (d)

Fig. 6. (a) First grasp in sequence which is an unstable grasp; (b) Posterior distribution after first grasp; (c) Second grasp, which is a stable grasp; (d) Posterior distribution after second grasp.

To test the framework, we displaced the object 40 mm, while setting $\mu_o = (0, 0)$ and $\sigma_o = (40, 4)$. The results of the two grasps executed with these parameters are presented in Figure 6. As can be seen from the figure, the posterior distribution converges close to the real pose in X. However, θ remains largely uncertain, because the measurement is not able to disambiguate this. The stability probabilities reflect reality well as after the first grasp, the probability of achieving a stable grasp given by the model is only 2%, which in reality was an unstable grasp, but after the second grasp the probability has grown to 63%, which was a stable grasp when lifting the object.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a framework for grasping, which operates in a probabilistic setting. The framework allows grasp planning, measurements, and corrective motions to interact, leading to a system where the uncertainty about the environment can be decreased simultaneously while planning and executing statistically optimal grasps. We also presented a demonstration of our framework utilizing MCMC methods. The demonstration showed that the approach is able to plan a stable grasp and simultaneously update the pose estimate of the object. The models used in the demonstration can also be extended, for example, the model for stability, $P(S|G, O)$, can be extended to include tactile information, $P(S|G, O, T)$, which has been done in previous work [1]. We believe that adding input from tactile sensors can further benefit the pose and grasp stability estimation as, for example, we can measure contact surface types and shapes.

To extend the framework, ongoing work is focused on applying the framework to more complex objects and statistical models. We also intend to focus on determining object identity and category using our framework in addition to the pose estimation shown in this paper. Our plans include working both with simulations for the repeatability and real hardware. Finally, it should be mentioned that the framework is applicable in any manipulation scenario where a success function, attempted actions, sensor models and environment variables can be defined.

REFERENCES

[1] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 616 –629, 2011.

[2] V. Christopoulos and P. Schrater, "Handling shape and contact location uncertainty in grasping two-dimensional planar objects," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Nov. 2007, pp. 1557 –1563.

[3] C. Corcoran and R. Platt, "A measurement model for tracking hand-object state during dexterous manipulation," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 4302 –4308.

[4] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2342–2348.

[5] C. Ferrari and J. Canny, "Planning optimal grasps," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, May 1992, pp. 2290 –2295 vol.3.

[6] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 1628 –1633.

[7] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp Planning Via Decomposition Trees," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 4679–4684.

[8] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia Grasp Database," in *IEEE International Conference on Robotics and Automation*, 2009.

[9] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-Driven Grasping with Partial Sensor Data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[10] R. Detry, D. Kraft, A. Buch, N. Kruger, and J. Piater, "Refining grasp affordance models by experience," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2287 –2293.

[11] C. Barck-Holst, M. Ralph, F. Holmar, and D. Kragic, "Learning grasping affordance using probabilistic and ontological approaches," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, June 2009, pp. 1 –6.

[12] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct. 2010, pp. 1579 –1585.

[13] R. Platt, "Learning grasp strategies composed of contact relative motions," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, Dec. 2007, pp. 49 –56.

[14] K. Goldberg and M. Mason, "Bayesian grasping," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, May 1990, pp. 1264 –1269 vol.2.

[15] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Task-driven tactile exploration," in *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[16] A. Petrovskaya, S. Thrun, D. Koller, and O. Khatib, "Guaranteed inference for global state estimation in human environments," in *RSS 2010 Mobile Manipulation Workshop*, 2010.

[17] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 569 –585, june 2011.

[18] J. Laaksonen and V. Kyrki, "Probabilistic approach to sensor-based grasping," in *ICRA 2011 Workshop on Manipulation Under Uncertainty*, 2011.

[19] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[20] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robot. Automat. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[21] R. Koiva, R. Haschke, and H. Ritter, "Development of an intelligent object for grasp and manipulation research," in *The 15th International Conference on Advanced Robotics*, 2011, pp. 1285–1291.

[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.

Contact detection and location from robot and object tracking on RGB-D images

Author Names Omitted for Anonymous Review. Paper-ID [132]

***Abstract*—In this paper, we address the problem of detecting contacts between a robot hand and an object during the approach and execution phases of manipulation tasks in the absence of touch perception. In order to detect contact in such conditions, we implemented a method which uses the visual tracking of objects using ICP (Iterative Closest Point) to detect small movements of the object's RGB-D point-cloud which has been previously segmented. Once a movement has been detected, a combination of a 3D occupancy grid of the object and an sphere-based model of the robot is used to probabilistically estimate the locations of contacts. The proposed approach is implemented and experimentally validated for several relevant cases.**

I. Introduction

The robot manipulation and grasping of objects on service scenarios is a complex task as a consequence of the uncertainty which appears in such conditions. The limited or complete absence of knowledge about the objects to be manipulated, the limited accuracy of the sensors used to perceive the scene, and the difficult calibration of the kinematics of complex manipulators are the most relevant difficulties. Manipulation planners and controllers must take into account for these uncertainties in their design and implementation.

In such grasping systems, the ability to perceive and detect the contact between the robot and the objects becomes a critical feature to ensure the security and robustness of the execution. The detection of contacts or, in a more broad definition, the sense of touch have been achieved by integrating in the robot hands and arms a variety of contact and proximity sensors. It is not unusual that the most advanced robotic hands include some type of contact sensors. This is the case of the *Shadow Dexterous Hand* [14], the *DLR-HIT-Hand* [10] and many others.

Several technological approaches have been developed to provide the sense of touch. Three main categories can be distinguished depending on their configuration and characteristics. The first one is tactile sensing which aims to imitate the sensibility of the human skin [15, 3]. It usually consists of a single or an array of cells placed on the surface of a body, which measure the existence of contact and the pressure or related magnitudes in such locations. The main advantage of these sensors is that they are able to determine the location of the contact accurately [5].

A second category is composed by sensors that measure strain or force/torque. They are devices which interface two different bodies and measure the forces and torques transmitted between them. When a contact occurs on one of the bodies, it produces a force or torque that is transmitted to the rest of the connected bodies and thus can be detected by the sensor[12, 13]. These sensors can be considered global contact sensors since they are potentially able of detecting any contact on a body, but can hardly identify accurately the location and magnitude of the contact forces involved. Finally, the third category is composed by proximity sensors. Although they are not properly contact sensors, they have been used to identify imminent contacts and thus used in a similar fashion as touch sensors [9]. Their main advantage is that they can be used without perturbing the state of the objects.

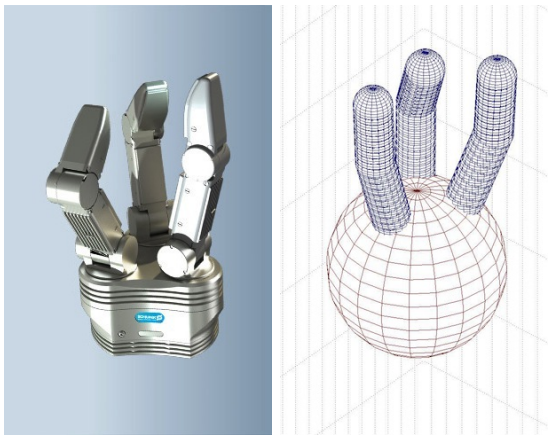


Fig. 1. Shunk Dextrous Hand: real and spherical model

Although these approaches have extensively been used, they are not free of limitations. Tactile sensors are unable to detect contacts in surfaces which are not covered by them. Force-torque sensors cannot detect locations of contacts and their sensitivity is low when the contact forces are small. In addition, all these sensors require a wiring and a system integration which causes their implementation difficult. As a consequence, even in heavily sensorized hand-arm systems, it is common that a contact goes unnoticed which compromises the stability of the grasping and manipulation actions.

This paper describes a novel approach which uses vision to detect and analyse contacts. It is based on the basic assumption that if an object moves is because it has been touched. We present a system that simultaneously tracks the robot actuator and the object on the scene. When a movement on the object is detected, it analyses the occupancy information of the object and the robot to infer the probable location of the contact in the surface of the robot.

(a) s-tope with two spheres, bi-sphere

(b) s-tope with tree spheres, tri-sphere

Fig. 2. Examples of simple s-topes: bi-spheres and tri-spheres

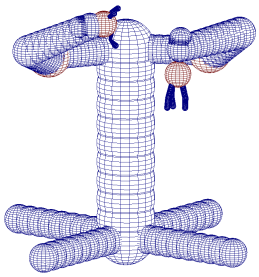


Fig. 3. Spherical model of our robotic system.

II. Description of the system and underlying assumptions

Our approach assumes an scenario in which a robot system is composed of at least a robot arm and a hand. Its purpose is to approach the hand and manipulate a single rigid object lying on a planar surface. No previous model of the object is available and no assumptions about its shape or aspect are made.

The only sensor modality that our approach is going to use is a *Kinect* sensor which delivers RGB-D images and point clouds describing the scene. Other sensor modalities like tactile sensors are used exclusively for validation purposes in the experimental section (see Sec. VII). Assumptions are made that the kinematics of the arm and hand are known and the correspondence between coordinates of the scene point-cloud and the frame of the robot arm are calibrated accurately.

III. Geometrical Modelling of Robot System

In our approach, we use a geometric model of the robot system to reason about the space occupied by the robot and estimate contacts with objects, especially when the robot is occluded in the RGB-D image.

We have chosen a model based in bounding volume primitives to describe our robot system, in particular we choose the spherically extended polytopes, *s-topes*, as bounding volumes. This representation has been widely used [17, 4, 6] because of their efficiency in distance computation, specifically in collision detection and path planning. An *s-tope* [7] is the convex hull of a finite set of spheres $s \equiv (c,r)$, where c is the center and r is its radius. Given the set of n spheres $S = \{s_0, s_1, ..., s_n\}$, the convex hull of such a set, S_s, contains an infinite set of swept spheres expressed by Eq. 1 .

$$S_s = \left\{ s : s = s_0 + \sum_{i=0}^{n} \lambda_i (s_i - s_0), s_i \in S, \lambda_i \geq 0, \sum_{i=0}^{n} \lambda_i \leq 1 \right\} \quad (1)$$

Where λ_i is the parameter that determines a specific sphere, radius and center, of the whole set of spheres. To illustrate the previous equation, Figure 2 depicts several examples of s-topes defined by two (*bi-spheres*) and three spheres (*tri-spheres*).

We have modeled our robot as a combination of s-topes. Each link is represented as a bi-sphere and some static parts as singles spheres. In addition, each defining sphere has been attached to the corresponding frame of the kinematic chain. Figure 3 depicts the complete model of our robot manipulator system and Figure 1 illustrates a detail with the model of our three-fingered robot gripper.

IV. Object Segmentation

This section describes the identification and the segmentation of the target object in the scene. As it has been described previously, we assume that the scene contains a single object, lying on a planar surface. The scene is obtained by a RGB-D camera that provides a 3D point cloud. This point cloud contains points which belongs to the object, the supporting table and the robot manipulator. It is necessary a procedure to segment the object and isolate the 3D points belonging to it.

The fist step is to transform all points in the point cloud from the camera frame to the robot base frame. Then, several filters are applied to remove points from the point cloud that do not belong to the object. The first one removes the points belonging to the supporting plane. It consists of a predefined frame box which discards all the points outside the box. The dimensions of this box has been calibrated to fit exactly the dimensions of the table in our scenario. This filter keeps all the points in the scene that are over the table, containing the object and the robot manipulator.

The second filter uses the spherical model of the robot to determine which points in the point cloud belong to the robot. A point is considered to belong to the robot if the distance between it and the model is less than zero. This distance is the minimum from the point to all the s-topes which compose the robot model. Since our geometric model is composed only of spheres and bi-spheres, we need to apply only two rules to compute each distance. In the cases of a single sphere, the distance between a point p_i and the sphere $s_i \equiv (c_i, r_i)$ is computed using Eq.2, where c_i is the centre and r_i the radius of the sphere:

$$distance = \| \overrightarrow{p_i} - \overrightarrow{c_{min}} \| - r_{min} \quad (2)$$

In the case of the distance between a point p_i and a bi-sphere, we first need to determine the closest sphere to the point among the infinite number which define the bi-sphere. Given a bi-shpere defined by the spheres $s_1 \equiv (c_1, r_1)$ and $s_2 \equiv (c_2, r_2)$, Eq. 3 defines the rule to find the closest sphere

$s_{min} \equiv (c_{min}, r_{min})$ to p_i. Then, Eq. 2 can be used to compute the distance.

$$\begin{aligned} \lambda_{min} &= -\frac{(\overrightarrow{c_1} - \overrightarrow{p_i}) \cdot (\overrightarrow{c_2} - \overrightarrow{c_1})}{\|\overrightarrow{c_2} - \overrightarrow{c_1}\|^2}; \quad \lambda_{min} \in [0,1] \\ \overrightarrow{c_{min}} &= \overrightarrow{p_i} - \overrightarrow{c_1} + \lambda_{min}(\overrightarrow{c_2} - \overrightarrow{c_1}) \\ \overrightarrow{r_{min}} &= \overrightarrow{p_i} - \overrightarrow{r_1} + \lambda_{min}(\overrightarrow{r_2} - \overrightarrow{r_1}) \end{aligned} \tag{3}$$

All the remaining points in the 3D point cloud are evaluated and those which distance in zero or negative are labelled as belonging to the robot and then removed from the point cloud. After this filter, the remaining points are considered to be part of the object.

To illustrate the process of object segmentation, Figure 4 shows results after each step of the process. Picture 4(a) shows the initial image of the scene. Figure 4(b)) shows the points remaining after the box filtering. Finally, the third image on the right (Fig. 4(c)) shows the object segmented from the whole scene. This process is repeated each time step in order to calculate the position of the object.

V. Contact detection

Once an object has been segmented from the scene, the next step is to determined when the object moves. As our approach is constrained to rigid solids and non articulated objects, we assume that when an object moves is because a contact has occurred. In order to detect the movement, it is necessary to track the point cloud of the object using consecutive images.

Literature offers algorithms for object tracking based on image descriptors such as SIFT [11], SURF [2], or Harris corners [8]. These methods make the assumption that the objects have a texture, a regular shape or that the descriptors are visible all the time. As we do not have any assumption of shape or texture and the hand of the robot may occlude parts of the object, we have chosen the Iterative Closest Point (ICP) algorithm [18] as our approach to object tracking. In particular, we use the standard ICP algorithm implemented in PCL library [1]. The ICP does not need any characteristic of the object, only two points clouds are needed. The ICP refines iteratively the transform between two consecutive point clouds by repeatedly generating pairs of corresponding points on the meshes and minimizing an error metric. Once the object is segmented as was described in Section IV, the ICP is applied to the object point cloud at instant t^k and t^{k-1}. The algorithm returns a homogeneous transformation matrix that is decomposed in a translation vector $t(x,y,z)$ and a quaternion $q(x,y,z,w)$ from which an angular rotation ω is obtained . When the module of the translation vector $\|\overrightarrow{t}\|$ or the angular rotation ω are grater than a threshold ($\|\overrightarrow{t}\| \geq t_{max}$ or $\omega \geq \omega_{max}$), we conclude that the object has moved and therefore has been contacted by the hand.

VI. Estimation of contact location

This section describes how the location of the detected contact is estimated. The output is a point cloud containing all the points with high likelihood of being in contact with the object. Contact location estimation from vision is going to find the problem of occlusion, either because the object is occluded by the object or because the robot itself occludes the object. In this sections we describe a procedure that deals with point contact occlusions and gives a guess about where the contact point is.

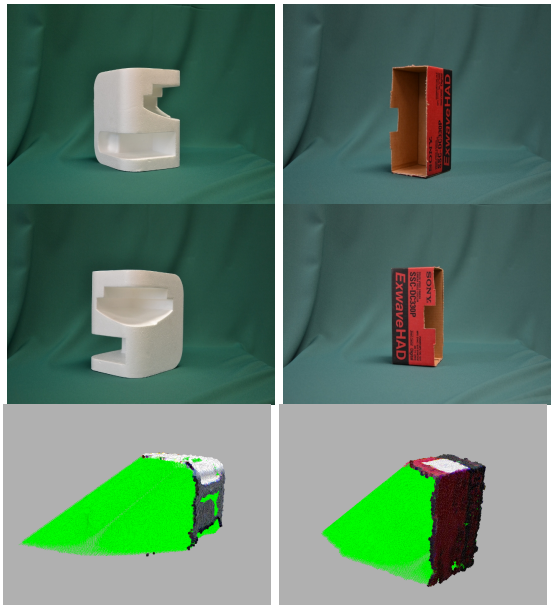


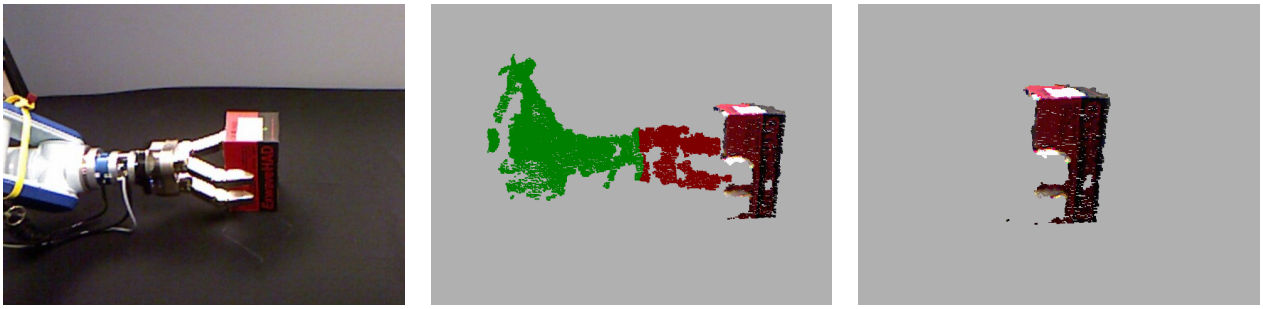
Fig. 5. Real objects and its initial Occupancy Grid Map (OGM).

With the purpose of dealing with the uncertainty of the occluded area we use an Occupancy Grid Map (OGM) that is iteratively updated to estimate which space areas have the greatest probability to be in contact. Initially an OGM of the object is built assuming that the object is not occluded by the hand (Fig. 5). While the hand is moving towards the object to perform any task, the OGM is updated exploiting the robot movement. If a contact is detected (see Sec.V) the intersection between the hand model surface and the OGM is built. For each point of the intersected region its probability of being a contact point is calculated. Section VI-B tells the details for the generation and updating of the OGM and section VI-B1 describes how the contact point likelihood is calculated.

A. Occupancy Grid Map Initialization

The OGM is bulit projecting each point of the initial object point cloud along the direction of the camera until its intersection with the table plane. To project the points, a perfect pin-hole model of the camera is used. We also assume the table plane position to be known in advance.

Then, the occluded area is discretized in cells of $1mm^3$ in each direction (x,y,z). The cells that appear already in the point cloud (i.e. are being seen by the camera) have a probability of being occupied $P(c_i = occ) = 1$. Meanwhile there is no information about the cells that are not being seen its starting likelihood of being occupied is $P(c_i = occ) = 0.5$.



(a) Scene Example

(b) Table points removed; arm (green) and hand (red) points labelled

(c) object points

Fig. 4. Object segmentation process

B. Occupancy Grid Map Update

In the traditional occupancy map the cell values are updated with a distribution function of the measurement sensor. However no direct sensor to measure contact is considered in this approach. Instead, we propose to use the hand model as a sensor.

1) Virtual Contact Sensor: In order to delimit the area where the contact is, we have taken into account the direction of movement of the hand $\overrightarrow{u}$. Firstly, the cells of the OGM that belong to the surface of the hand are obtained. Secondly, for each cell, the normal vector to the hand model surface $\overrightarrow{n}$ is computed and the angle α between the normal vector $\overrightarrow{n}$ and the movement vector $\overrightarrow{u}$ is obtained. Finally, if this angle is less than an empirically defined treshold α_{max} the point is considered part of the sensible area and a candidate to be in contact. Fig.6 shows the result of a simulated contact with different hand movement directions and α_{max}, where red points are contact candidates (i.e. have an $\alpha \leq \alpha_{max}$) and blue points are not contact candidates (i.e. points with an $\alpha \geq \alpha_{max}$).

In order to put the above mentioned in a mathematical way we have formulated it in eq. 4. This equation shows the probability function, $P(z(k)|c_i)$, of a given cell to have a contact. This function depends on the distance between the cell and the robot spherical model. If the cell is inside the model $P(z(k)|c_i) = 0$, we have the guarantee that the cell is free of contact. In the case that the cell is outside of the hand model there is no way to have any new information about the occupancy, thus $P(z(k)|c_i) = 0.5$.

Finally if the cell is on the surface of the hand model and $\alpha \leq \alpha_{max}$ a probability function is defined, where $d \in [0,5]mm$, is the variable that models the error in the direction of hand movement. s is the standard deviation of the Gaussian used to take into account the error in the geometrical and kinematic model of the hand. In other words, we consider that the contact surface estimation has a Gaussian error in the direction of the movement.

$$P(z(k)|c_i) = \begin{cases} 0 & p \in \text{inside model} \\ 0.15 \cdot exp(\frac{d^2}{2 \cdot s^2}) + 0.5 & p \in \text{surface}; \alpha \leq \alpha_{min} \\ 0.5 & otherwise \end{cases} \quad (4)$$

2) Update Algorithm: The OGM is updated each time step, depending on the contact detection (Sec. V). If no contact is detected, the cells that are inside or in the surface of the model are updated. The new likelihood value this cells is 0, as can be obtained from eq. 4 and eq. 5. Cells that are outside the model and do not belong to the surface, keep the same value. Finally, if contact has been detected then the surface cells are updated with the eq 5:

$$P(c_i = occ|z(k))^k = \frac{P(c_i = occ|z(k))^{k-1} \cdot P(z(k)|c_i = occ)}{\sum_{c_i} (P(c_i|z(k))^{k-1} \cdot P(z(k)|c_i))} \quad (5)$$

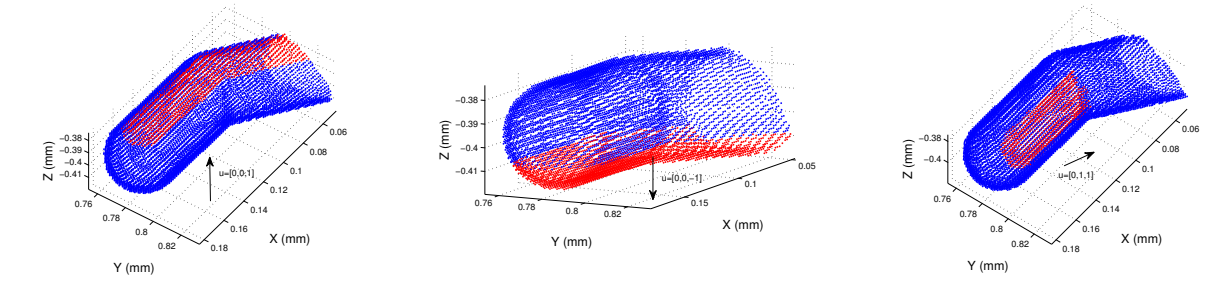
where $P(z(k)|c_i = occ)$ is the probability of a measure given a occupied cell ($c_i = occ$), in other words, the sensor model. $P(c_i = occ|z(k))^{k-1}$ is a priori probability to be occupied given a measure $z(k)$ and $P(c_i = occ|z(k))^k$ is the a posteriori probability to be occupied given a measure $z(k)$, for more information refer to book [16].

When the contact is detected, our algorithm returns a point cloud with the cell that have a high probability level (e.g more than 0.8) and are part of the sensible surface as has been described in section VI-B1. The fig. 7 shows all the surface points, the candidate points that belong to the sensible surface are colored from blue to red depending on its contact probability.

The inputs of the alg. 1 are the object point cloud, the joint value of arm and hand, the initialized grid map, and the direction of movement of the hand. The function in line 7 is the detection of contact that was described in section V.

VII. Experimental Validation

In order to validate our approach, we have implemented the algorithms for our robot platform and tested them on



(a) $\alpha_{max} = 0.6rad$, $u = 0,0,1$ (b) $\alpha_{max} = 1rad$, $u = 0,0,-1$ (c) $\alpha_{max} = 0.6rad$, $u = 0,1,1$

(d) $\alpha_{max} = 1rad$, $u = 0,1,-1$ (e) $\alpha_{max} = 1rad$, $u = -1,0,0$ (f) $\alpha_{max} = 1.6rad$, $u = -1,0,0$

Fig. 6. Sensor surface with different values of α_{max} and hand direction u

Algorithm 1 Grid Map Update

$ObjectpointCloud \leftarrow INPUT$
$q_{arm} \leftarrow INPUT$ {Arm Joints, Hand Joints}
$Grip\_Map \leftarrow INPUT$
$\overrightarrow{u}_{hand} \leftarrow INPUT$ {Hand movement direction}
$d \in [0,5]$
$distance \in$ {inside, surface, outside}
$contact \leftarrow Contact\_Detection$
for all $cell_i$ **do**
  $[distance, \overrightarrow{n}_{surf}] \leftarrow DistanceToModel(cell_i, q_{arm}, q_{hand})$
  $\alpha = \frac{acos(\overrightarrow{n}_{surf} \cdot \overrightarrow{u}_{hand})}{\|\overrightarrow{n}_{surf}\| \cdot \|\overrightarrow{u}_{hand}\|}$
  if $cell_i \in surface$ **and** contact **then**
    for all d **do**
      $UpdateCellSurface(cell_i, d, \overrightarrow{u}_{hand}, Grid\_Map)$
    end for
    $ContactSurface \leftarrow cell_i$
  else
    if $cell_i \in inside$ **then**
      $cell_i = 0$
    end if
  end if
end for
if contact **then**
  return ContactSurface
end if

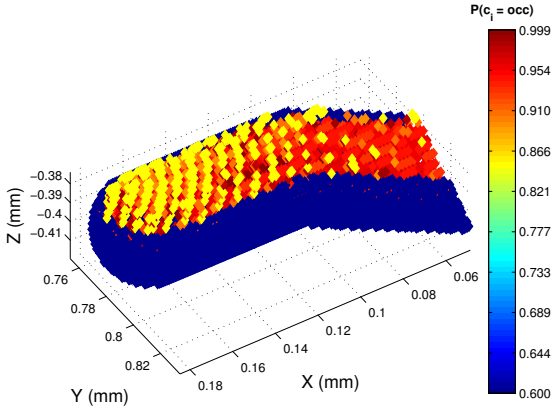


Fig. 7. contact probability distribution

Algorithm 2 $UpdateCellSurface(cell_i, d, u_{hand}, Grid\_Map)$

$s = 0.01$
$\overrightarrow{cell}_i = GetCoodinateFromCell(Grid\_Map, cell_i)$
$\overrightarrow{cell}_d = \overrightarrow{cell}_i + d \cdot \overrightarrow{u}_{hand}$
$P(c_i = occ|z(k))^{k-1} = GetCellValue(Grid\_Map, \overrightarrow{cell}_d)$
$P(z(k)|c_i = occ) = 0.15 \cdot exp(\frac{d^2}{2 \cdot s^2}) + 0.5$
$Pc_i = occ|z(k))^k =$
$= \frac{P(c_i=occ|z(k))^{k-1} \cdot P(z(k)|c_i=occ)}{(P(c_i|z(k))^{k-1} \cdot P(z(k)|c_i)) + (1-P(c_i|z(k))^{k-1}) \cdot (1-P(z(k)|c_i))}$
return $Pc_i = occ|z(k))^k$

several cases. Our robot platform is an upper body humanoid composed of two 7 DOF robot arms, a Schunk Dextrous Hand with 7 DOF, equipped with a JR3 Force-Torque sensor on the wrist. The hand has three fingers, each with 2 DOF and two Weiss tactile sensors: one on the fingertip and one on the inner phalanx (see Fig. 1). The vision system is composed of a 2 DOF pan-tilt head with a *Kinect* sensor. For our experiments, the force-torque sensor has not been used and the tactile sensors have been only used to provide ground truth validation data.

Two validation experiments have been implemented. The first one seeks to demonstrate that our approach is more sensitive than the real tactile sensors. The second one evaluates our approach for estimating contacts with occluded parts of non convex objects. The objects used for validation are shown in Fig. 5. The object on the left is a white very light object, with a non-trivial shape. The one on the right is an empty light cardboard box.

A. Contact detection sensibility

The first experiments aims to demonstrate that our approach is able to detect contacts that our real tactile sensors can not detect. One single object is in the scene and the hand configuration and approaching movement are set in such a way that the first contact with the object occurs on the fingertip of the middle finger, where there is a tactile sensor. The controller is programmed so that as soon as a tactile contact is detected, the hand stops. The difficulty here is that the object is too light and, in most of the cases, the contacts go unnoticed due to the low sensibility of the real tactile sensors.

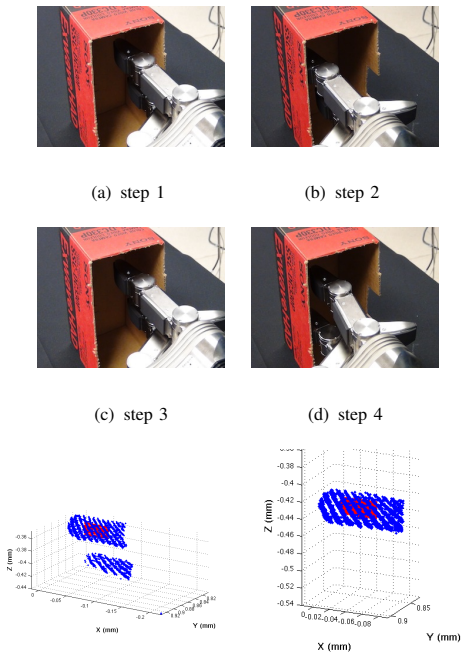
The initial configuration is shown in the upper rows of figures 8(a), 8(b), 8(c) and 8(d). The robot moves in a predefined direction towards the object. When the contact is detected, the robot stops and the contact information is stored. The results of the experiments using the proposed approach are shown in middle rows of figures 8(a) and 8(c) where as soon as a small movement has been observed on the object the hand stopped. Lower rows of figures 8(a) and 8(c)) show a detail of the estimated contact locations on the fingertips.

Middle and lower rows of figures 8(b), 8(d) show the case where the visual contact detection was disabled so the robot relies exclusively on real tactile sensor information. In both cases the contact was not detected and the hand kept pushing the object.

B. Validation of the estimation of contact locations

In this set of experiments, we seek to test the validity of our approach to estimate the locations of contacts. In order to validate this, we compared the output of our estimator against the readings given by the real tactile sensors. As the objects are too light to be robustly detected by the tactile sensors, we have loaded the objects with additional weights.

The experiments consist on a single object placed on the table. A set of different hand configurations and directions are predefined to approach the object. The robot moves until a visual contact is detected. Then, the visually estimated contacts locations and the real tactile sensor readings are recorded.



(a) step 1 (b) step 2

(c) step 3 (d) step 4

(e) Movement to avoid false contacts

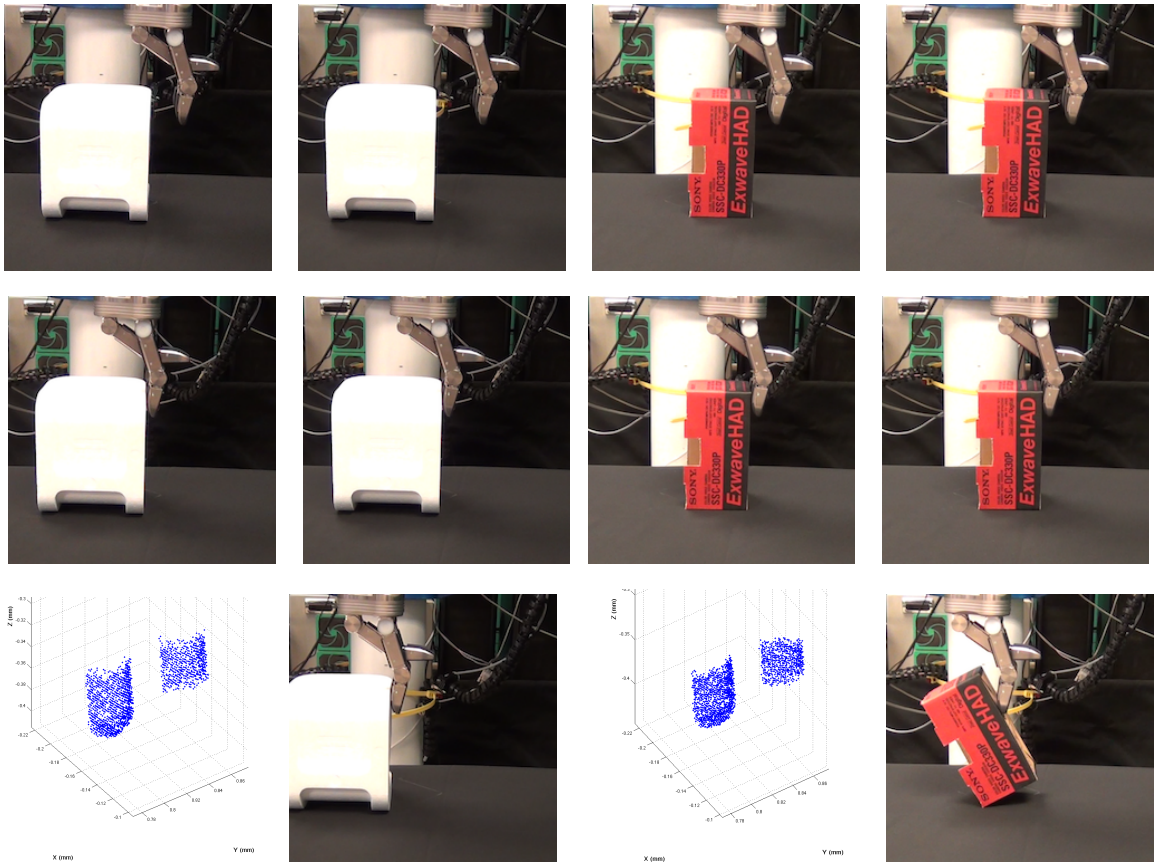
Fig. 9. Improvement of the estimation

Figure 10 shows the results of these experiments. The graphs in the right of each figure show the estimated contact locations (in blue) and real contact tactile readings (in red). The 3D positions of the tactile readings are reconstructed using the known kinematics and configuration of the robot. As it can be seen, the proposed estimator is able to detect accurately the contact locations. A special case is shown in the fifth row of fig. 10 in which not all the occluded robot fingers contact the object, as the readings of the real tactile sensor indicate. However, the visual contact estimator shows that the contact is equally probable on both fingertips. The reason is that when the contact is detected the cells of the occupancy grid map around the finger are unknown, so their probability of being occupied is 0.5.

These type of ambiguities can be solved by further exploration movements of the fingers. An example of this is shown in Figure 9, in which a second movement is performed to get a second contact detection. Figure 9(e) shows the initial estimation (left) and the estimation after the second movement (right) which match with the real sensor readings.

VIII. Conclusion and Further Work

This paper has described a contact sensor which uses exclusively RGB-D images, without the help of any touch sensors. The whole process is divided in several steps. On the first step the object point cloud is segmented from the whole scene using a spherical geometric modelling of the robot. On the second step, the object point cloud is tracked using an



(a) Contact estimation object 1 (b) Real tactile sensor object 1 (c) Contact estamimation object 2 (d) Real tactile sensor object 2

Fig. 8. Sensibility comparative between contact estimation vs real

ICP approach in order to detect any relevant movement. As soon as a movement is detected a probability estimator is used to update the occupancy grid of the object and to detect the most likely locations where the contact has happened. This approach is validated through several experiments on a real scenario where the sensibility of the visual contact sensor to detect contacts that would be unnoticed by real sensors is demonstrated. In addition, the ability of the sensor to detect contacts when the hand is visually occluded is also demonstrated.

The paper presents a first implementation of a contact sensor based exclusively on visual information. This has never been demonstrated to the knowledge of the authors. Regarding the utility of such a sensor, it is not minded to be used alone, but in combination with other touch modalities, in order to obtain a more robust contact detections and, as consequence, a more reliable and robust manipulation of objects. The experiments have shown that the visual sensors can provide information that touch sensors are unable, i.e.: when the robot contacts light objects, or when the contact happens on not-sensorized surfaces.

Fianlly, more technically, the methodology can still be improved. First, methods to speed-up the computations are required. ICP is specially time-consuming and faster alternatives would be necessary to make the approach work at an acceptable frame ratio. Another important improvement is to design schemes to integrate visual and touch modalities of contact detections, in order to design more robust manipulation controllers.

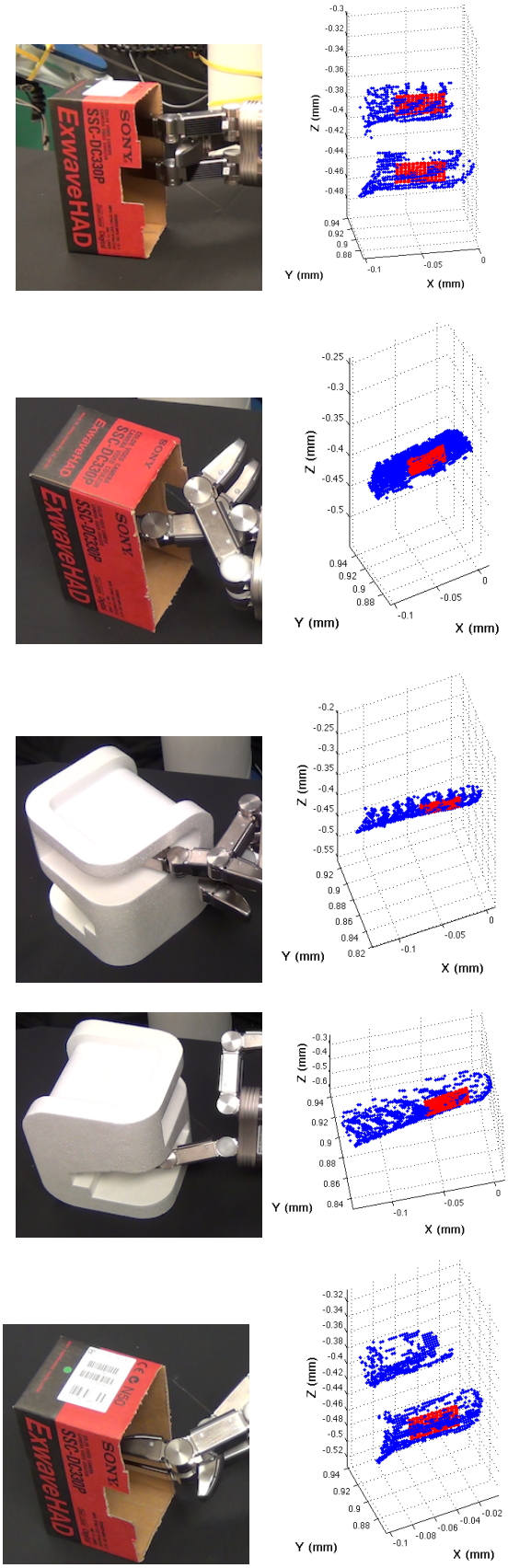
Finally, some secondary parts of the algorithm could also be improved in order to make their use more general. In particular the segmentation phase in the case that the scene contains several objects, stacked objects or articulated bodies.

REFERENCES

[1] Point cloud library (PCL). URL http://pointclouds.org.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision Image Understanding*, 110:346–359, June 2008. ISSN 1077-3142.

[3] R.S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing - from humans to humanoids. *IEEE Transactions on Robotics,*, 26(1):1–20, feb. 2010.



(a) Real image

(b) Virtual sensor (blue) vs real sensor (red)

Fig. 10. Contact Estimation Validation

[4] A.P. del Pobil, M.A. Serna, and J. Llovet. A new representation for collision avoidance and detection. In *IEEE International Conference on Robotics and Automation*, pages 246 –251 vol.1, may 1992.

[5] J. Felip and A. Morales. Robust sensor-based grasp primitive for a three-finger robot hand. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1811 –1816, October 2009.

[6] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 4(2):193 –203, apr 1988.

[7] G.J. Hamlin, R.B. Kelley, and J. Tornero. Efficient distance calculation using the spherically-extended polytope (s-tope) model. In *IEEE International Conference on Robotics and Automation*, pages 2502 –2507 vol.3, may 1992.

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference*, pages 147–151, 1988.

[9] Kaijen Hsiao, Paul Nangeroni, Manfred Huber, Ashutosh Saxena, and Andrew Y Ng. Reactive grasping using optical proximity sensors. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2098–2105, 2009.

[10] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M.H. Jin, Y.W. Liu, R. Wei, and Z.W. Xie. The modular multisensory dlr-hit-hand. *Mechanism and Machine Theory*, 42(5):612 – 625, 2007.

[11] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150 –1157 vol.2, 1999.

[12] R. Platt, A.H. Fagg, and R.A. Grupen. Null-space grasp control: Theory and experiments. *IEEE Transactions on Robotics*, 26(2):282 –295, April 2010. ISSN 1552-3098.

[13] Mario Prats, Pedro J Sanz, and Angel P Pobil. A framework for compliant physical interaction. *Autonomous Robots*, 28(1):89–111, 2009.

[14] Shadow. URL http://www.shadowrobot.com/hand/.

[15] Johan Tegin and Jan Wikander. Tactile sensing in intelligent robotic manipulation a review. *Industrial Robot: An International Journal*, 32(1):64–70, 2005. ISSN 0143-991X.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. 2001.

[17] J. Tornero, J. Hamlin, and R.B. Kelley. Spherical-object representation and fast distance computation for robotic applications. In *International Conference on Robotics and Automation*, pages 1602 –1608 vol.2, apr 1991.

[18] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surface. *International Journal of Computer Vision*, 13(2):119–152, 1994.

Learning Continuous Grasp Stability for a Humanoid Robot Hand Based on Tactile Sensing

J. Schill and J. Laaksonen and M. Przybylski and V. Kyrki and T. Asfour and R. Dillmann

Abstract— **Grasp stability estimation with complex robots in environments with uncertainty is a major research challenge. Analytical measures such as force closure based grasp quality metrics are often impractical because tactile sensors are unable to measure contacts accurately enough especially in soft contact cases. Recently, an alternative approach of learning the stability based on examples has been proposed. Current approaches of stability learning analyze the tactile sensor readings only at the end of the grasp attempt, which makes them somewhat time consuming, because the grasp can be stable already earlier.**

In this paper, we propose an approach for grasp stability learning, which estimates the stability continuously during the grasp attempt. The approach is based on temporal filtering of a support vector machine classifier output. Experimental evaluation is performed on an anthropomorphic ARMAR-IIIb. The results demonstrate that the continuous estimation provides equal performance to the earlier approaches while reducing the time to reach a stable grasp significantly. Moreover, the results demonstrate for the first time that the learning based stability estimation can be used with a flexible, pneumatically actuated hand, in contrast to the rigid hands used in earlier works.

I. INTRODUCTION

Grasp stability in analytical sense is well defined and can be readily computed in simulation where enough data of the grasp is available, i.e. all contacts between the robotic hand and the object that is grasped. Additionally, using a force closure metric for grasp stability, one can compute a grasp that sufficiently resists outside forces, such as gravity, thus allowing the robot to manipulate the object, for example by lifting the object. However, when using real hardware, the tactile sensor data is imperfect, both in the sense of detecting contacts and in the sense of determining the actual contact forces. In some cases the proprioceptive information, i.e. joint configuration, is also difficult to determine accurately, thus, causing uncertainty in ascertaining the kinematic configuration of the hand. All these described phenomena pave a difficult road for computing the grasp stability analytically with real hands.

In this paper, we focus on learning the grasp stability instead of analytically solving it. Compared to the analytical methods, learning requires training data, which needs to be collected beforehand. As the training data, we can use any pertinent data that can be collected from robotic hand, in our case we use input from all tactile sensors and the hand finger configuration. It is also important to notice that the raw sensor data can be used in the learning, for example, there is no need to know the kinematic configuration of the hand to compute the true locations of the contacts when analytically solving the grasp stability. This feature allows grasp stability to be learned for many different robotic hands with only minimum changes.

There has been a number of publications on learning the grasp stability [1], [2]. These approaches evaluate the stability after the hand finished closing around an object. We extend the work presented in previous papers, so that the decision on the grasp stability can be achieved during the grasping instead of at the end of the grasp. We also demonstrate that the learning of the grasp stability is possible with the ARMAR-IIIb hand [3], [4] , a flexible anthropomorphic hand operating on pneumatics.

The rest of the paper is divided into four sections. Section II gives an overview on learning grasp stability as well as other learning approaches that are grasp and manipulation related. Section III introduces a theoretical background on machine learning methods and how they can be applied to the grasp stability problem. Section IV contains the experiments made on data collected using the ARMAR-IIIb hand. We conclude with discussion of the results in Section V.

II. RELATED WORK

Grasp stability analysis by analytical means is a well established field. However, to analytically determine the grasp stability, the kinematic configuration of the hand and the contacts between the hand and the object must be perfectly known. Previous studies on this subject are numerous and [5] gives a detailed review. However, the references are useful only in cases when conditions described above are true. When this is the case, it is possible to determine if the grasp is either force or form closure grasp [6], which ensures the stability. Compared to this body of work, we wish to learn the stability from existing data, i.e. the tactile data.

The research on use of tactile and other sensors in a grasping context has increased in last few years. Felip and Morales [7] developed a robust grasp primitive, which tries to find a suitable grasp for an unknown object after a few initial grasp attempts. However, only finger force sensors were used in the study.

Apart from using tactile information as a feedback for low level control [8], tactile sensors can be used to detect or

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215821.

J. Laaksonen and V. Kyrki are with Department of Information Technology, Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland, jalaakso@lut.fi, kyrki@lut.fi

M. Przybylski, J. Schill, T. Asfour and R. Dillmann are with the Humanoids and Intelligence Systems Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany, {markus.przybylski, schill, asfour, dillmann}@kit.edu

identify object properties. Jiméneza et al. [9] use the tactile sensor feedback to determine what kind of a surface the object has, which is then used to determine a suitable grasp for an object. Petrovskaya et al. [10] on the other hand use tactile information to reduce the uncertainty of the object pose, upon an initial contact with the object. In their work, a particle filter is used to estimate object's pose, but the tactile sensor used to detect contact with the object is not embedded in the gripper performing the grasping.

Object identification has been studied by Schneider et al. [11] and Schöpfer et al. [12]. Schneider et al. show that it is possible to identify an object using tactile sensors on a parallel jaw gripper. The approach is similar to object recognition from images and the object must be grasped several times before accurate recognition is achieved. Schöpfer et al. use a tactile sensor pad fixed to a probe instead of a gripped or hand. They also study on different temporal features which can be used to recognize objects. Similar object recognition systems have been presented in [13], [14].

The approach used and extended here has been published in [1]. Similar approach was used in [2]. However, in this paper we show that we can use described methods with a more complex hand, the ARMAR-IIIb humanoid hand, and that we can extend the single time instance classifier by means of filtering.

III. Supervised Learning of Grasp Stability

A. Learning Grasp Stability

Our notation of observations follows [1]:

- $D = [o_i], i = 1, \ldots, N$ denotes a data set with N observation sequences.
- $o_i = [x_t^i], t = 1, \ldots, T_i$ is an observation sequence with T_i samples.
- $x_t^i = [\mathbf{f}_t^i \ \mathbf{j}_t^i]$, each sample consists of $\mathbf{f}$, the features extracted from tactile sensors and $\mathbf{j}$, the joint configuration.

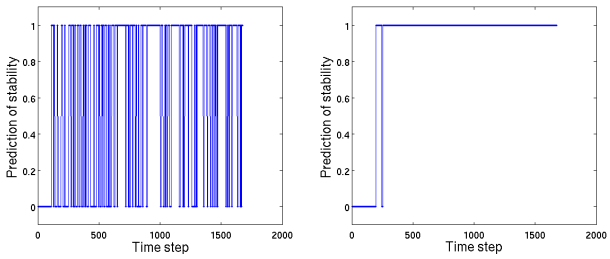
To learn grasp stability, the training data is collected from series of grasps, noted by the observation sequences o_i. Then, from each observation sequence the last sample, $x_{T_i}^i$, is used for the training. This captures the time instant on which the decision of stable or unstable grasp is based on. Both unstable and stable grasp must be included in the training data so that sufficient data is available to discern the stable grasps from the unstable grasps.

We use Support Vector Machine (SVM) [15] to classify the grasp as either stable or unstable. Compared to force closure metric from the analytical methods for computing the grasp stability, the binary classification is not as informative as the continuous value given by the force closure metric, however the classification result reflects the stability criteria in the training data directly. Another benefit of SVM is that it is computationally efficient, so that it can be used on-line during grasping.

B. Learning Temporal Changes in Grasp Stability

In [1], the temporal information collected during a grasp is used in conjunction with a hidden Markov model (HMM) to decide whether the grasp is stable or not. But for the method to be able to decide, the grasp must be completed. The second method presented in [1] was based on the idea depicted in III-A. We propose to extend the instantaneous SVM-based method by applying the learned stability model on-line to each sample $x_1, \ldots, x_T$ we obtain during the grasp, contrary to the previous approach, where only the final sample, x_T, is is used to determine the stability of the grasp. This extension allows quicker decision making on the grasp quality in the case of a stable grasp.

As the method described in III-A does not remember any of the previous time instances and does not consider the whole grasp sequence from $t = 1, \ldots, T$, the classification result over time may oscillate. One pathological example is shown Figure 1. Through the use of filtering and thresholding, the oscillations can be effectively removed.



(a) (b)

Fig. 1: (a): Each time instance of a stable grasp classified with a SVM classifier; (b): The classification result filtered with an exponential filter and thresholded.

We study two different filter types: a mean filter and an exponential filter. The results of the experiments with the filters are shown in Section IV. The input for the filters are the results from the classifier, either 0 or 1. The mean filter can be defined as a sliding window, with window size w. The mean of the data in the window is then calculated, and this result is the output of the filter. Exponential filter is described by

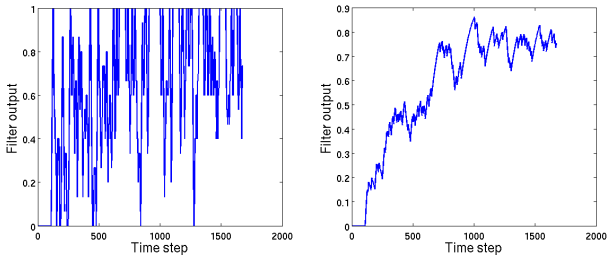
$$y(t) \ = \ (1 - \alpha) \cdot y(t-1) + \alpha \cdot x(t) \ . \qquad (1)$$

Equation 1 consists of $y(t)$ and $y(t-1)$, filter output at time instances t and $t-1$, of $x(t)$ the binary stability at time t and of α which a weighting factor. An examples of both filters are shown in Figure 2 which depicts the same sequence as in Figure 1.

Introducing the filters requires setting more parameters in addition to the parameters for SVM. These include w for the mean filter window width, and α for the exponential filter. In addition both require the threshold, thr, for the binary decision of stability. After the threshold has been crossed, the grasp is deemed stable. Close to optimal parameters can be found experiementally and we have done that for the datasets used in this paper.

In addition to the filters, we ran experiments without using any filters, thus, the output from the classifier is taken

directly. This approach provides a quicker response to stable grasps but can also misclassify unstable grasps as stable grasps more frequently than the filter based approach.



(a) (b)

Fig. 2: (a): Filter output of mean filter; (b): Filter output of exponential filter.

C. Feature Extraction

Each of the tactile sensors on the ARMAR-IIIb platform produces a tactile image. An example image showing all six tactile images is shown in Figure 3. This imaging property of the sensors allows us to use image feature extraction techniques. In this case we have chosen the image moments as our feature extractor, which have been shown to perform well in this task [16]. The hand comprises of two different sizes of tactile sensors which contain 4x7 or 4x6 tactile elements or taxels.

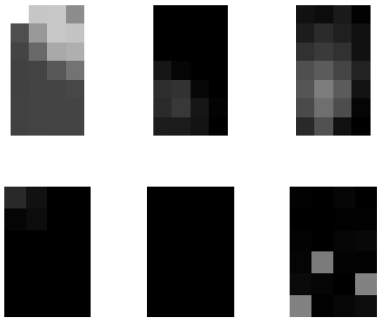


Fig. 3: Tactile images from ARMAR-IIIb.

Raw image moments are defined as

$$m_{p,q} = \sum_x \sum_y x^p y^q I(x,y) \ , \tag{2}$$

where $I(x,y)$ is the force measured by the taxel. The moments are computed up to order two, that is $(p+q) = o, \ \ o = \{0,1,2\}$. These are related to the total pressure, the mean of the contact area, and the shape of the contact area, indicated by the variance in x- and y-axes. Moments are computed for all tactile sensors individually, thus $\mathbf{f} \in \mathbb{R}^{36}$.

In addition to the tactile images, the joint angle sensors provide a source of information relevant to the stability of the grasp. However as the number of fingers and joints is usually much less than the number taxels (tactile sensing elements) in tactile sensors, it is reasonable to use the data from the joints directly. In this case, 8 joint angle sensors are available, thus $\mathbf{j} \in \mathbb{R}^8$. All feature vectors, x_t^i, were normalized to zero mean and unit standard deviation.

IV. EXPERIMENTS

A. Hardware Platform

We used the humanoid robot ARMAR-IIIb as a test platform for the experiments with our stability classifier. ARMAR-IIIb consists of several kinematic subsystems: The head, the torso, two arms, two hands, and the platform. The head has seven degrees of freedom (DoF) and contains four cameras, i.e. two cameras per eye. The torso has 1 DoF in the hip, allowing the robot to turn its upper body. Each of the two 7 DoF arms consists of a 3 DoF shoulder, a 2 DoF elbow and a 2 DoF wrist. At the tool center point (TCP) of each arm a FRH-4 Hand [17] is mounted. The hands are pneumatically actuated using fluidic actuators. For the experiments in this paper, we used ARMAR-IIIb's right hand, (see Fig. 4), which is equipped with joint encoders and pressure sensors. This allows a force position control of each DoF [18]. The hand has 1 DoF in the palm, and 2 DoF in the thumb, the index and the middle finger, respectively. Apart from that, there is 1 DoF for combined flexion of the pinky and ring finger. Furthermore the hand contains 6 tactile sensors from Weiss Robotics [19]. One tactile sensor is mounted on the distal phalanges of the thumb, the index and the middle finger, respectively. Three tactile sensors are mounted at the palm, in the area between the thumb and the index and middle fingers. The tactile sensors have a resolution of 4×7 taxel (phalanges) and 4×6 taxel (palm). They use a resisitive working principle to measure the pressure applied to the sensor. Therefore an array of electrodes is covered with a layer of conductive foam. When a pressure is applied to the sensor the resistance between the electrodes decreases, which is measured by an microcontroller. Further information can be found in [20], [21], [22].

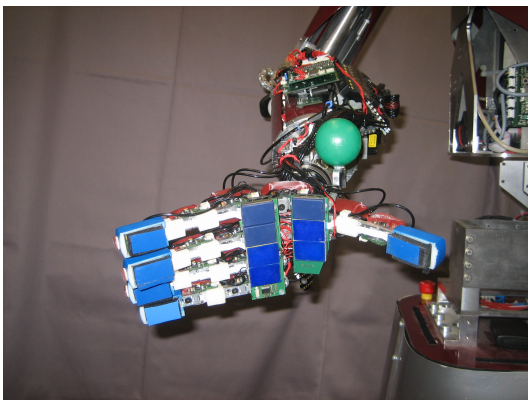


Fig. 4: ARMAR's right hand. Tactile sensors are mounted on the palm and the distal phalanges of the thumb, the index and the middle finger.

B. Data Collection

In order to provide sensor input for the training and the validation of the classifier, we needed to treat two distinct cases:

- Collect data for successful, stable grasps.
- Collect data for unstable grasps.

The second case also includes data for the cases where the hand cannot close completely or not at all,due to obstacles, and cases where the hand closes emptily, i.e. it does not experience contact to any object at all. Yet in all these cases one gets sensor readings that have to be considered for training and validating the classifier. We collected data from the following two types of sensors:

- Tactile sensor data
- Joint angle data of the hand joints

Fig. 5: The basket with our test objects.

For data collection, we executed grasps on a set of household items located in a box (see Fig. 5). The configuration of the objects in the box was modified between the individual test runs in order to allow the hand to reach a large variety of different end configurations. We used the following data collection procecure: First, we placed the box with the objects in front of the robot. Then we moved ARMAR's right hand to a pre-grasp pose near the target object. Different possible pre-grasp poses included the following:

- Grasps from the top where the hand would move vertically down.
- Grasps from the top, but with tilted approach directions.
- Grasps from the side.
- Varying roll angles of the hand with respect to the approach direction, for each of the three cases above.

After moving the hand to the pre-grasp pose, we started the data recording which means we began to read and store the tactile sensor data and joint angle data once during every pass of ARMAR's control loop. All data were labeled with a time stamp. In the next step, we moved the hand towards the object until the tactile sensors in the palm reported contact with the object. Then we closed the hand and waited until the pressure on the hand's actuator stabilized and would not grow anymore. The finger forces are set to the maximum to create a strong tactile image on the sensors. Due to the compliant characteristic of the hand, the hand adapts to the shape of the object. In this context we point out that we considered only three-fingered grasps, i.e. we only closed the thumb, the index and the middle finger during grasping. We did not close the ring and small finger, as they are not equipped with tactile sensors and thus they would not contribute to the tactile sensor input of the classifier. After closing the hand, we stopped the recording of the sensor data. Finally, we tried to lift the object by moving ARMAR's hand up. Then, we reported the result of the experiment, i.e. whether the grasp was successful or not. We repeated the above procedure until enough samples had been collected. We collected two separate sets, D_1 and D_2. D_1 contained 71 stable grasps and 94 unstable grasps. D_2 comprised of 82 stable grasps and 76 unstable grasps. By collecting two separate sets with different grasps, we can get an idea of the generalization capability of the classifier which was tested in the validation tests. Figures 6 and 7 show some successful grasps from the validation tests. The left column shows the situation after closing the hand. The right column shows the grasps after lifting the respective object.

TABLE I: Confusion matrix for classification rates of grasps when classifying only the last sample, for datasets D_1 and D_2.

D_1	P. Stab.	P. Unstab.	D_2	P. Stab.	P. Unstab.
Stable	0.79	0.21	Stable	0.72	0.28
Unstable	0.28	0.72	Unstable	0.26	0.74

C. Experimental Results

We have divided the experiments into two parts. The first part consists of synthetic tests, which presents the reliability and accuracy of the classification of the grasp stability and comparisons between different filter types. The second part is validation test, using a learned stability model with the real ARMAR-IIIb platform.

1) Synthetic tests: In the synthetic tests, we used both datasets D_1 and D_2. For most experiments, confusion matrix is presented, showing how the classifier performs in terms of true positives (stable, predicted stable), false positives (unstable, p. stab.), true negatives (unstable, p. unstab.) and false negatives (stable, p. unstab.).

In Table I, the SVM was trained with data from corresponding dataset, only the last sample from each observation sequence was classified, to enable comparison to earlier works. The reported results are averages from 10-fold cross validation. The results show that the performance across datasets is similar. These results can be compared with reported results in [1], [2], showing that the ARMAR-IIIb hardware is able to reach similar performance as the Schunk Dextrous Hand (SDH) or the Barrett hand in this task.

Contrary to results in Table I, in Tables II, III and IV the whole observation sequence was classified using the methodology presented in Section III-B. In Table II, the mean filter was used with window width of 25 and with threshold

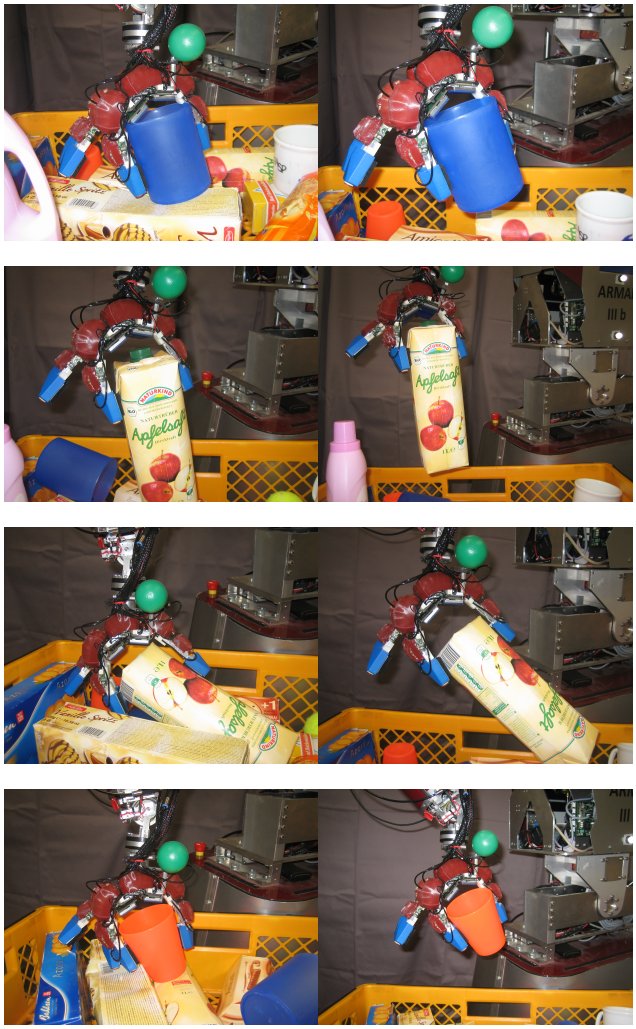


Fig. 6: Some example grasps. Left column: situation immediately after closing the hand. Right column: After lifting the object.

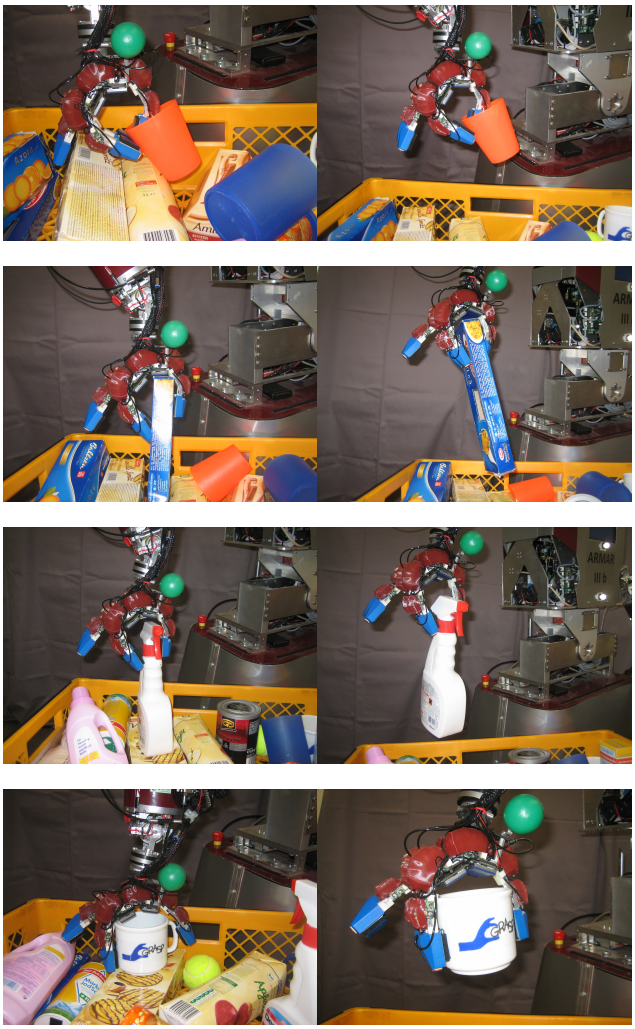


Fig. 7: Some example grasps. Left column: situation immediately after closing the hand. Right column: After lifting the object.

of 0.61, Table III shows result with an exponential filter with $\alpha = 0.056$ and threshold of 0.61. These parameter values were searched for using grid search and produced the best results for both datasets. Results in Table IV were obtained without using a filter.

Overall, when using a filter with the classification, the overall classification rate is similar to the last sample classification, but classification rate of the unstable grasps is better. This can be explained through the use of the filter which filters out the effect of the last sample, thus, leading to a better classification result. In the case where no filters are used, in Table IV, the stable grasps are predicted well, but this translates also to falsely predicting that unstable grasps are stable. On average, the filter based classification is better in predicting the stable and unstable grasps across the two datasets.

One interesting possibility that comes with the method described in Section III-B is that the grasp sequence can be

TABLE II: Confusion matrices for classification rates of grasps using mean filter ($w = 25$, thr $= 0.61$).

D_1	P. Stab.	P. Unstab.	D_2	P. Stab.	P. Unstab.
Stable	0.77	0.23	Stable	0.74	0.26
Unstable	0.24	0.76	Unstable	0.16	0.84

TABLE III: Confusion matrices for classification rates of grasps using exponential filter ($\alpha = 0.056$, thr $= 0.61$).

D_1	P. Stab.	P. Unstab.	D_2	P. Stab.	P. Unstab.
Stable	0.79	0.21	Stable	0.73	0.27
Unstable	0.23	0.77	Unstable	0.16	0.84

TABLE IV: Confusion matrices for classification rates of grasps without a filter.

D_1	P. Stab.	P. Unstab.	D_2	P. Stab.	P. Unstab.
Stable	0.90	0.10	Stable	0.87	0.13
Unstable	0.46	0.54	Unstable	0.21	0.79

TABLE V: Average percentage of time steps to reach decision on stable grasp compared to full grasp observation sequence.

D_1	Mean filt.	Exp. filt.	No filt.
Time	68.6%	66.9%	59.6%

TABLE VI: Confusion matrices for validation tests.

Mean filt.	P. Stable	P. Unstable
Stable	0.77	0.23
Unstable	0.39	0.61
Exp. filt.	P. Stable	P. Unstable
Stable	0.76	0.24
Unstable	0.38	0.62
No filter	P. Stable	P. Unstable
Stable	0.90	0.10
Unstable	0.46	0.54

stopped when the classifier decides that a stable grasp has been achieved. Table V presents the results with different filter types. For example, if a whole grasp sequence is 1000 time steps long, the classification using a mean filter can stop the grasp at time step 686 on average, if the grasp is a stable grasp. Without a filter, the average time goes down as expected but with a cost of overall classification rate as seen in Table IV.

2) Validation tests: To mimic a real world usage scenario, dataset D_1 was used to train the SVM classifier. Then using the trained classifier, dataset D_2 was classified. Each observation sequence in the dataset was classified with mean and exponential filters and without filtering. The results are show in Table VI. Compared to results in Table I, the number of false positives rises. This effect might be due to tactile sensor hysteresis, i.e. the output from the sensors changes between the collection of datasets which in turn means that dataset D_1 does not represent the data in D_2 and leads to worse results.

V. CONCLUSIONS

In this paper, we focused on learning grasp stability from labeled data, similar to approaches in [1], [2]. We utilized a well-known classifier, SVM, and trained it using grasp data acquired from the sensors of the humanoid hand of ARMAR-IIIb. We showed that we are able to reach similar results with ARMAR-IIIb as previously reported on other types of hardware, such as Schunk Dextrous Hand or Barrett hand. We also extended the SVM based grasp stability classifier with use of filters to whole grasp sequence instead of just the end of the grasp sequence. This allows faster decisions for stable grasps.

REFERENCES

[1] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 616 –629, 2011.

[2] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 5917 –5922.

[3] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control," in *IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, December 2006, pp. 169–175.

[4] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward Humanoid Manipulation in Human-Centred Environments," *Robotics and Autonomous Systems*, vol. 56, pp. 54–65, January 2008.

[5] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *ICRA*, 2000, pp. 707–714.

[6] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, 1st ed., B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer-Verlag, 2008.

[7] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *IEEE/RSJ International. Conference on Intelligent Robots and Systems*, Oct. 2009.

[8] T. Tsuboi and et al., "Adaptive grasping by multi fingered hand with tactile sensor based on robust force and position control," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 264–271.

[9] A. Jiménez, A. Soembagijo, D. Reynaerts, H. V. Brussel, R. Ceres, and J. Pons, "Featureless classification of tactile contacts in a gripper using neural networks," *Sensors and Actuators A: Physical*, vol. 62, no. 1-3, pp. 488–491, 1997.

[10] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *ICRA*, 2006, pp. 707–714.

[11] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *In Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2009.

[12] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," in *14th International Conference on Advanced Robotics*, ser. Proceedings of the ICAR 2009, IEEE. Munich, Germany: IEEE, 22/06/2009 2009.

[13] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010, pp. 2342–2348.

[14] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, "Haptic object recognition using passive joints and haptic key features," in *In Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[15] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[16] J. Laaksonen, V. Kyrki, and D. Kragic, "Evaluation of feature representation and machine learning methods in grasp stability learning," in *10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 112–117.

[17] I. Gaiser, S. Schulz, A. Kargov, H. Klosek, A. Bierbaum, C. Pylatiuk, R. Oberle, T. Werner, T. Asfour, G. Bretthauer, and R. Dillmann, "A new anthropomorphic robotic hand," in *IEEE-RAS International Conference on Humanoid Robots*, Daejeon, Korea, 2008.

[18] A. Bierbaum, J. Schill, T. Asfour, and R. Dillmann, "Force Position Control for a Pneumatic Anthropomorphic Hand," in *IEEE-RAS International Conference on Humanoid Robots*, Paris, France, 2009, pp. 21 – 27.

[19] Weiss Robotics, "Tactile sensor module, type: DSA 9335," accessed 10/09/2009. [Online]. Available: http://www.weiss-robotics.de/

[20] D. Göger, N. Gorges, and H. Wörn, "Tactile Sensing for an Anthropomorphic Robotic Hand: Hardware and Signal Processing," in *Proc. of the IEEE Int. Conf. on Robotics and Automation, May 12 - 17, 2009, Kobe, Japan*, 2009.

[21] D. Göger and H. Wörn, "A highly versatile and robust tactile sensing system," in *Proc. of the IEEE Conf. on Sensors*, Atlanta (GA), USA, 2007.

[22] K. Weiss and H. Wörn, "The working principle of resistive tactile sensor cells," in *Proc. of the IEEE Int. Conf. on Mechatronics and Automation, Canada*, 2005.