EU FP7 CogX
ICT-215181
May 1 2008 (52months)

DR 2.4:
Manipulation of previously unseen objects

Michael Zillich, Sebastian Zurek, Marek Kopicki, Rustam Stolkin, Yasemin Bekiroglu, Renaud Detry

TUW, Vienna
⟨zillich@acin.tuwien.ac.at⟩

Due date of deliverable:	July 31, 2011
Actual submission date:	July 29, 2011
Lead partner:	TUW
Revision:	final
Dissemination level:	PU

This deliverable reports work related to object manipulation. While work reported in previous periods to a large degree relied on pre-trained models, the work reported here relaxes these dependencies on known models in two ways. First, our work on learning forward models to predict the movements of pushed objects using a combination of multiple experts is now able to generalise beyond known shapes and manipulative actions and was shown to work on real object manipulations. Moreover these learned predictors are used as motion model within a visual object tracker, showing increased robustness in visually challenging situations. Second, our work on assessing grasp stability based on tactile feedback is able to generalise learned grasp stability predictors to novel objects. This allows execution of stable grasps for previously unseen objects and was also shown to improve simulation based grasp planning for known objects.

Executive Summary

This report combines two pieces of work on (a) self extending modular motor learning and (b) grasping of previously unseen objects:

Regarding (a) this report presents work on motor learning, in which a combination of multiple experts is able to learn forward models to predict the movements of objects subjected to manipulative actions. The report also shows how the system can make useful predictions about previously unseen objects, by making use of knowledge acquired by observing the motions of other objects. This report further presents the use of learned predictors as motion model of a particle filter based tracker, showing improved robustness in visually challenging situations.

Regarding (b) this report presents work related to predicting and monitoring grasp stability. Grasping of previously unseen objects is a very useful ability for learning new objects, which is one major means of extending a robot's knowledge in CogX. Lacking detailed object models to plan a stable grasp haptic feedback becomes a valuable source of information.

Self extending modular motor learning

The motor learning work extends previous work [21, 20] in which a system learned probability distributions of how a rigid body would move when subjected to manipulative pushing actions by a robot arm. The report explains how additional predictive experts can be trained which encode information about the relative motions of surface patches of the object with respect to parts of the environment, such as a supporting ground plane. These additional experts enable a greater degree of generalisation, i.e. making predictions about the motions of previously unseen objects with novel shapes, and predicting the motion of objects in response to previously unexplored manipulative actions. We have now shown that we can learn forward models capable of such generalisation for real object manipulations.

Building on the work on model based tracking reported in DR.2.1 and DR.2.2 [36, 43, 35] we extended our particle filter based tracking with a motion model, which uses the learned predictors of object motion. This is especially useful in visually challenging situations like toppling objects, where fast motion and sudden velocity changes would overwhelm simpler (e.g. first order) motion models.

The work presented here led to two conference publications [23, 34] and one recent journal submission [22].

Role of self extending modular motor learning in CogX

This work relates to Task 2.5 - Modular motor learning theory because it addresses the problem of learning about contact relations, and predicting object trajectories of manipulated objects.

In order to fill the knowledge gap that results from the discovery of a novel object, the agent should be able to learn about how that object will behave when subjected to manipulative actions. Furthermore, it is advantageous if the agent can make use of its previous experience with known objects, to make useful predictions about how a novel object will behave.

Contribution to the CogX scenarios and prototypes

This work contributes to the Dexter scenario, which deals with learning about physical object properties via active experimentation.

Grasping of previously unseen objects

This report presents means of monitoring the stability of a grasp applied onto a novel object, using the tactile signals captured by sensors placed on the robot's fingers. As a secondary contribution, this report presents means of learning object-specific tactile characterisations from interactions with novel objects, in order to facilitate further use of these objects. This work was evaluated on the KTH manipulation platform (industrial arm and dexterous hand). It has led to one journal [4] and two conference publications [2, 1]. The work presented here builds on the contributions presented in DR 2.3 [3].

Role of grasping of previously unseen objects in CogX

The ability to grasp novel objects is paramount in learning about them. The tasks defined in CogX often confront the agent with an object which it does not know about. In order to fill the knowledge gap that results from the discovery of a novel object, the agent must learn the object's appearance, its physical properties, and its usefulness and purpose. To this end, the agent must be able to interact with the object, and, for instance, grasp the object and move it in front of a camera to capture the object's 3D appearance.

To the end of allowing the agent to grasp previously unseen objects, we addressed the problem of monitoring the stability of a grasp applied onto a novel object, using the tactile signals captured by sensors placed on the robot's fingers. Grasps planned onto novel objects will often be uncertain, as the information available to the planner before the grasp is limited. Monitoring the stability of a grasp from tactile feedback allows the agent to abort a grasp that is unlikely to succeed, and it potentially prevents damaging the object or the robot.

Contribution to the CogX scenarios and prototypes

This work mainly contributes to the Dexter scenario, and in particular to DR 7.2, which is concerned with manipulation under partial information.

1 Tasks, objectives, results

1.1 Self extending modular motor learning

1.1.1 Planned work

This deliverable reports work related to Task 2.6:

> Task 2.6: Self-extending modular motor learning. Create a modular motor learner that adds new contexts to its model, based on assessing the quality of its predictions. We will use a probabilistic model to identify whether to refine existing modules, or add a new module, and to control exploration while doing so. (M22 - M32)

In period 2 we developed a basic theory of modular prediction, involving products of experts, and also an extension using a mixture of experts for context identification. In period 3 we planned to deliver several pieces of work. The first was to show how the products theory applies in real robot cases, and in particular to show how we can perform extrapolative rather than interpolative generalisation, i.e. generalisation of predictions to previously unseen shapes and actions. This kind of self-extension is extremely challenging. The second was how our mixture model learner should explore an object most effectively so as to identify a context. Initial work on extracting qualitative states has also been undertaken, and will be reported in the correct deliverable at month 50.

1.1.2 Actual work performed

Annexes 2.1 and 2.2 present our new work on the product of experts model. In this work a variety of expert predictors are trained on different kinds of information about the behaviour of pushed objects. In this period we showed how the learned predictors can be combined to generalise extrapolatively rather than interpolatively as is normal in machine learning. This extrapolative generalisation is an exceptionally hard problem. In these papers we show that extrapolative generalisation works well when the learner has good information about the object trajectory when it is learning. Given good trajectory information the products model can successfully generalise predictions to new actions outside of the hull of previously tried actions. The same model can also extrapolatively generalise with respect to shape, learning on a cylinder and a box and generalising to a pair of linked cylinders, or learning on a polyflap and generalising to a box.

One problem in learning the above predictors from observed visual tracks is that visual trackers can lose track of the object in unexpected situations such as an object suddenly toppling over. Unfortunately it is precisely these situations that provide valuable training input. We solved this problem by running a particle filter based tracker [36, 35] with a very high number of particles, thus increasing its robustness while sacrificing real time performance.

While this is acceptable for an offline learning system, it is not suitable for a robot autonomously extending its knowledge in real time. To this end we investigated methods to improve tracking robustness especially in such visually challenging situations by using the learned predictors themselves as the motion model within the tracker. Simple (e.g. first order) motion models for recursive filters have been used in the past, but these can not capture the sudden changes of motion resulting from pushing or toppling an object. The learned predictors allow the tracker to overcome situations such as toppling objects with a combination of fast motion and image blur or complete occlusions during motion. Most importantly this paves the way for a bootstrapping system, where a partially learned predictor starts to improve visual tracking, which in turn allows improved observation of learning situations and so on, which is the intended next step in this line of work. This work led to a publication at ICRA 2011 (see Annex 2.3).

1.1.3 Relation to the state-of-the-art

Push manipulation is particularly interesting in that pushes can give rise to a large number of unstable poses in 3D rigid bodies. However, most previous work on push manipulation in robots is restricted to planar sliding motions of what are effectively 2D objects [28, 27, 39, 9]. There is little literature addressing the more complex problem of push manipulations on real 3D bodies, which are free to tip or roll. It is possible to use physics simulators to predict the motion of interacting rigid bodies. However, this approach is reliant on explicit knowledge of the objects, the environment, and key physical parameters which can be difficult to tune. Even then, such predictions may not be possible due to inherent limitations of the physical model employed, for example when modeling friction.

Machine learning approaches have been developed to learn to classify or provide predictions for objects or object classes, e.g. rolling versus non-rolling objects[14, 44], or liftable versus non-liftable objects [38]. These kinds of approach are limited, in that predictions learned may not be generalisable to a new object, pose or push direction, and explicit 6-DOF rigid body motions are not predicted. In contrast, our approach learns to make predictions of explicit 3D rigid body transformations. The probabilistic nature of the learning enables generalisation to novel push directions, object poses, and objects with novel shapes.

Recent approaches to visual tracking aim at improving robustness in tough real-world scenarios by using robust filtering techniques [18, 19], combining different types of features (such as edges and interest points) [29, 48], taking advantage of fast parallel GPU architectures [10, 30, 37, 26, 45, 36, 12, 8] or combinations of these. The work presented in this report falls into the above categories, but replaces the simple motion model (static or first order) often used in these trackers with a complex nonlinear motion model provided by learned predictions of object motion under manipulation.

1.2 Grasping of previously unseen objects

1.2.1 Planned work

This deliverable reports work related to Task 2.8:

> Task 2.8: Grasping novel objects. Based on our object models, we will investigate the scalability of the system with respect to grasping novel, previously unseen objects. We will demonstrate how the system can execute tasks that involve grasping based on the extracted sensory input (both about the scene and individual objects) and taking into account its embodiment. (M27 - M50)

Task 2.8 spans the second half of the project. Grasping novel objects requires (1) the ability to plan grasps for novel objects, and (2) the ability to execute the planned grasps robustly. This report addresses the second point: robustly executing grasps planned onto novel objects.

1.2.2 Actual work performed

The first contribution in this section is an agent that learns and memorises what it “feels” like to grasp objects. As a result of this learning process, the agent is able to predict, from tactile feedback obtained early during grasp executions, whether a grasp is going to be stable or unstable. The agent initially trains itself by performing several grasps on various objects. After each grasp, the agent lifts up the object and turns it upside-down. If the object stays rigidly bound to the hand during this movement, the grasp is marked as successful. During training, the agent encounters both successful and unsuccessful grasps, which provide it with input-output pairs, in the form of tactile imprint streams (input) and success/failure labels (output). These data are used to train a classifier that predicts success probabilities from the continuous stream of tactile signals that are received while the agent closes the robot’s hand around an object. We demonstrated that the acquired experience allows the agent to robustly predict grasp success when manipulating the objects used for training, and also when attempting to grasp previously-unseen objects. This work led to a publication in the high-impact journal *IEEE Transactions on Robotics* (See Annex 2.4).

The second contribution in this section (Annex 2.5) is a robot grasping system that combines a simulation-based grasp planner with the tactile-based stability predictor discussed in the paragraph above. Simulation-based grasp planners allow agents to plan grasps onto objects for which no manipulation experience exists, by exploiting planning algorithms that rely only on models of object shape. However, these planners usually overlook many important object properties, such as friction or mass distribution. Moreover, plans can never be executed perfectly due to uncertainty in perceptual input (e.g., noise in a vision-based computation of an object's pose). The grasps suggested by simulation-based planners are thus often of uncertain practical usability. By combining a simulation-based planner with the tactile-based stability model, the agent is able to estimate, before lifting up an object, whether the object-gripper contacts that were actually achieved are likely to lead to a stable grasp. If the tactile information does not predict a stable grasp, the agent can adjust its grasping configuration or retract its manipulator and try a new grasping plan.

The third contribution in this section (Annex 2.6) is a grasp stability predictor that exploits both tactile and visual information. We integrated our tactile models with the pose tracker developed at TUW, to create an agent that is able to learn what objects should feel like *when grasped from a specific side*. Stability estimates are based on both tactile imprints and the object-relative gripper pose read before and until the robot's manipulator is fully closed around an object. By comparing these models to the models defined on tactile perceptions or pose information alone, we demonstrated that joint tactile and pose-based perceptions carry valuable grasp-related information, as models trained on both hand poses and tactile parameters perform better than the models trained exclusively on one modality. We note that, because our models rely on the pose of an object, each model that the agent learns is only usable with that particular object. To overcome this limitation, we propose to learn models that characterise only a *part* of an object, and which would thus be applicable to novel objects that share the same part.

1.2.3 Relation to the state-of-the-art

Most of the work on grasp stability assessment relies on analytical methods [13, 31, 7, 50]. Compared with our approach, the analytical methods used by many grasp planners to estimate the stability of a grasp require exact knowledge of the contacts between the hand and the object, but that knowledge is usually uncertain in unstructured environments.

Tactile sensing has been used for various purposes in prior studies. The focus of the studies has been on the use of tactile data for object manipulation control [33, 40, 25, 42], exploration of object properties such as pose [41], surface type [17], shape [6] and deformation properties [11] or object recognition [47]. In our study, the main difference is that the tactile sensors are used to assess the stability of a grasp before further manipulating the object.

Vision-driven grasping and manipulation have been extensively studied [49, 24, 46, 32]. Vision has typically been used to plan grasping actions, and to update action parameters as objects move. Touch-based grasp controllers have also been studied, with emphasis on designing programs for controlling finger forces to avoid slippage and to prevent crushing objects [5, 16, 15]. To our knowledge, assessing grasp success by learning to differentiate between successful and unsuccessful grasping configurations jointly from live visual and tactile feedback has not been attempted before.

2 Annexes

2.1 Kopicki et al. “Learning to predict how rigid objects behave under simple manipulation”

Bibliography Kopicki, Marek; Zurek, Sebastian; Stolkin, Rustam; Morwald, Thomas; Wyatt, Jeremy : “Learning to predict how rigid objects behave under simple manipulation”, Proc. Int. Conf. Robotics and Automation (ICRA), pages 5722–5729, 2011

Abstract An important problem in robotic manipulation is the ability to predict how objects behave under manipulative actions. This ability is necessary to allow planning of object manipulations. Physics simulators can be used to do this, but they model many kinds of object interaction poorly. An alternative is to learn a motion model for objects by interacting with them. In this paper we address the problem of learning to predict the interactions of rigid bodies in a probabilistic framework, and demonstrate the results in the domain of robotic push manipulation. A robot arm applies random pushes to various objects and observes the resulting motion with a vision system. The relationship between push actions and object motions is learned, and enables the robot to predict the motions that will result from new pushes. The learning does not make explicit use of physics knowledge, or any pre-coded physical constraints, nor is it even restricted to domains which obey any particular rules of physics. We use regression to learn efficiently how to predict the gross motion of a particular object. We further show how different density functions can encode different kinds of information about the behaviour of interacting objects. By combining these as a product of densities, we show how learned predictors can cope with a degree of generalisation to previously unencountered object shapes, subjected to previously unencountered push directions. Performance is evaluated through a combination of virtual experiments in a physics simulator, and real experiments.

Relation to WP This work relates to Task 2.5 because it addresses the problem of learning about contact relations, and predicting object trajectories of manipulated objects. The learned models not only apply to objects encountered during training but also show generalisation to novel shapes and actions.

2.2 Kopicki et al. "Learning forward models for the motion of manipulated objects"

Bibliography Kopicki, Marek; Zurek, Sebastian; Stolkin, Rustam; Morwald, Thomas; Wyatt, Jeremy : "Learning forward models for the motion of manipulated objects", Submitted to IEEE Trans. Robotics, 2011.

Abstract An important problem in robotic manipulation is the ability to predict how objects behave under manipulative actions. This ability is necessary to allow planning of object manipulations. Physics simulators can be used to do this, but they model many kinds of object interaction poorly. An alternative is to learn a motion model for objects by interacting with them. In this paper we address the problem of learning to predict the interactions of rigid bodies in a probabilistic framework, and demonstrate the results in the domain of robotic push manipulation. A robot arm applies random pushes to various objects and observes the resulting motion with a vision system. The relationship between push actions and object motions is learned, and enables the robot to predict the motions that will result from new pushes. The learning does not make explicit use of physics knowledge, or any pre-coded physical constraints, nor is it even restricted to domains which obey any particular rules of physics. We use regression to learn efficiently how to predict the gross motion of a particular object. We further show how different density functions can encode different kinds of information about the behaviour of interacting objects. By combining these as a product of densities, we show how learned predictors can cope with a degree of generalisation to previously unencountered object shapes, subjected to previously unencountered push directions. Performance is evaluated through a combination of virtual experiments in a physics simulator, and real experiments.

Relation to WP This work relates to Task 2.5 because it addresses the problem of learning about contact relations, and predicting object trajectories of manipulated objects. This work shows the utility of our frameowrk on real data for extrapolation to other shapes and actions.

2.3 Mörwald et al. "Predicting the Unobservable: Visual 3D Tracking with a Probabilistic Motion Model"

Bibliography Mörwald, Thomas; Kopicki, Marek; Stolkin, Rustam; Wyatt, Jeremy; Zurek, Sebastian; Zillich, Michael; Vincze, Markus: "Predicting the Unobservable: Visual 3D Tracking with a Probabilistic Motion Model", Proc. Int. Conf. Robotics and Automation (ICRA), pages 1849–1855, 2011

Abstract Visual tracking of an object can provide a powerful source of feedback information during complex robotic manipulation operations, especially those in which there may be uncertainty about which new object pose may result from a planned manipulative action. At the same time, robotic manipulation can provide a challenging environment for visual tracking, with occlusions of the object by other objects or by the robot itself, and sudden changes in object pose that may be accompanied by motion blur. Recursive filtering techniques use motion models for predictor-corrector tracking, but the simple models typically used often fail to adequately predict the complex motions of manipulated objects. We show how statistical machine learning techniques can be used to train sophisticated motion predictors, which incorporate additional information by being conditioned on the planned manipulative action being executed. We then show how these learned predictors can be used to propagate the particles of a particle filter from one predictor-corrector step to the next, enabling a visual tracking algorithm to maintain plausible hypotheses about the location of an object, even during severe occlusion and other difficult conditions. We demonstrate the approach in the context of robotic push manipulation, where a 5-axis robot arm equipped with a rigid finger applies a series of pushes to an object, while it is tracked by a vision algorithm using a single camera.

Relation to WP This work is related to Task 2.6: Self-extending modular motor learning. While the work reported in Sec. 2.1 uses a visual tracker to provide training data for learning to predict object motion, this paper goes a step further and closes the loop, using the learned predictor as motion model for the visual tracker. This significantly improves tracking robustness, especially in visually difficult occlusion or toppling situations, where simpler (e.g. linear) motion models would fail. Although not fully exploited in the above paper, this paves the way for more autonmous learning, where tracking improved by better predictions, and prediction improved by better tracking can bootstrap each other.

2.4 Bekiroglu et al. “Assessing Grasp Stability Based on Learning and Haptic Data”

Bibliography Bekiroglu, Yasemin; Laaksonen, Janne; Jørgensen, Jimmy Alison; Kyrki, Ville; Kragic, Danica : “Assessing Grasp Stability Based on Learning and Haptic Data”, IEEE Transactions on Robotics, 27(3): 616–629, 2011

Abstract An important ability of a robot that interacts with the environment and manipulates objects is to deal with the uncertainty in sensory data. Sensory information is necessary to, for example, perform online assessment of grasp stability. We present methods to assess grasp stability based on haptic data and machine-learning methods, including AdaBoost, support vector machines (SVMs), and hidden Markov models (HMMs). In particular, we study the effect of different sensory streams to grasp stability. This includes object information such as shape; grasp information such as approach vector; tactile measurements from fingertips; and joint configuration of the hand. Sensory knowledge affects the success of the grasping process both in the planning stage (before a grasp is executed) and during the execution of the grasp (closed-loop online control). In this paper, we study both of these aspects. We propose a probabilistic learning framework to assess grasp stability and demonstrate that knowledge about grasp stability can be inferred using information from tactile sensors. Experiments on both simulated and real data are shown. The results indicate that the idea to exploit the learning approach is applicable in realistic scenarios, which opens a number of interesting venues for the future research.

Relation to WP This work is concerned with stability assessment in robotic grasping. We developed a method which allows a robot to learn what stable grasps feel like. Based on tactile input extracted during a grasp, the robot can detect when a grasp feels unstable, and act accordingly. We showed that these tactile models can provide useful information for grasping previously unseen objects.

2.5 Bekiroglu et al. "Integrating grasp planning with online stability assessment using tactile sensing"

Bibliography Bekiroglu, Yasemin; Huebner, Kai; Kragic, Danica : "Integrating grasp planning with online stability assessment using tactile sensing", IEEE International Conference on Robotics and Automation, pages 4750–4755, 2011.

Abstract This paper presents an integration of grasp planning and online grasp stability assessment based on tactile data. We show how the uncertainty in grasp execution posterior to grasp planning can be dealt with using tactile sensing and machine learning techniques. The majority of the state-of-the-art grasp planners demonstrate impressive results in simulation. However, these results are mostly based on perfect scene/object knowledge allowing for analytical measures to be employed. It is questionable how well these measures can be used in realistic scenarios where the information about the object and robot hand may be incomplete and/or uncertain. Thus, tactile and force-torque sensory information is necessary for successful online grasp stability assessment. We show how a grasp planner can be integrated with a probabilistic technique for grasp stability assessment in order to improve the hypotheses about suitable grasps on different types of objects. Experimental evaluation with a three-fingered robot hand equipped with tactile array sensors shows the feasibility and strength of the integrated approach.

Relation to WP This paper exploits our previous work on tactile learning to robustly execute grasps planned in an analytic grasp simulator. Simulator-based grasp planners have the advantage of being applicable to any object for which a model is available. Unfortunately, object models will often not hold *all* the properties that influence how an object and the robot's embodiment interact. In this work, local object-embodiment relations are captured by learning a tactile-based grasp stability model. Combining this model to a grasp planner allows the robot to robustly plan and execute grasps on a variety of objects.

2.6 Bekiroglu et al. "Learning Tactile Characterizations of Object- and Pose-specific Grasps"

Bibliography Bekiroglu, Yasemin; Detry, Renaud; Kragic, Danica : "Learning Tactile Characterizations of Object- and Pose-specific Grasps", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011

Abstract Our aim is to predict the stability of a grasp from the perceptions available to a robot before attempting to lift up and transport an object. The percepts we consider consist of the tactile imprints and the object-gripper configuration read before and until the robot's manipulator is fully closed around an object. Our robot is equipped with multiple tactile sensing arrays and it is able to track the pose of an object during the application of a grasp. We present a kernel-logistic-regression model of pose- and touch-conditional grasp success probability which we train on grasp data collected by letting the robot experience the effect on tactile and visual signals of grasps suggested by a teacher, and letting the robot verify which grasps can be used to rigidly control the object. We consider models defined on several subspaces of our input data – e.g., using tactile perceptions or pose information only. Our experiment demonstrates that joint tactile and pose-based perceptions carry valuable grasp-related information, as models trained on both hand poses and tactile parameters perform better than the models trained exclusively on one perceptual input.

Relation to WP This paper addresses the problem of learning a grasp model from both visual and tactile information. The agent learns grasp models for novel objects from experience, by exploring grasping configurations around canonical hand configurations demonstrated by a human. Compared to our previous work on tactile learning, this work makes use of vision-based object pose data to further discriminate between stable and unstable grasps.

References

[1] Yasemin Bekiroglu, Renaud Detry, and Danica Kragic. Learning Tactile Characterizations of Object- and Pose-specific Grasps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[2] Yasemin Bekiroglu, Kai Huebner, and Danica Kragic. Integrating grasp planning with online stability assessment using tactile sensing. In *IEEE International Conference on Robotics and Automation*, pages 4750–4755, 2011.

[3] Yasemin Bekiroglu, Danica Kragic, and Ville Kyrki. Learning grasp stability based on tactile data and hmms. In *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN), Viareggio, Italy*, 2010.

[4] Yasemin Bekiroglu, Janne Laaksonen, Jimmy Alison Jørgensen, Ville Kyrki, and Danica Kragic. Assessing Grasp Stability Based on Learning and Haptic Data. *IEEE Transactions on Robotics*, 27(3):616–629, 2011.

[5] A. Bicchi, J.K. Salisbury, and P. Dario. Augmentation of Grasp Robustness Using Intrinsic Tactile Sensing. In *IEEE International Conference on Robotics and Automation*, pages 302–307, 1989.

[6] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A Potential Field Approach to Dexterous Tactile Exploration. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2008.

[7] C. Borst, M. Fischer, and G. Hirzinger. Grasp Planning: How to Choose a Suitable Task Wrench Space. In *IEEE International Conference on Robotics and Automation*, pages 319–325, 2004.

[8] J. Brown and D. Capson. A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle Filter. *IEEE Transactions on Visualization and Computer Graphics*, PP(99), 2011.

[9] D. J. Cappelleri, J. Fink, B. Mukundakrishnan, V. Kumar, and J. C. Trinkle. Designing Open-Loop Plans for Planar Micro-Manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 637–642, 2006.

[10] J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade. GPU-Accelerated Real-Time 3D Tracking for Humanoid Locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[11] 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*, pages 2342–2348, 2010.

[12] Changhyun Choi and H.I. Christensen. Real-Time 3D Model-Based Tracking Using Edge and Keypoint Features for Robotic Manipulation. In *IEEE Int. Conf. on Robotics and Automation*, pages 4048–4055, 2010.

[13] C. Ferrari and J. Canny. Planning Optimal Grasps. In *IEEE International Conference on Robotics and Automation*, pages 2290–2295, 1992.

[14] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning about Objects through Action – Initial Steps Towards Artificial Cognition. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 3140–3145, 2003.

[15] R.D. Howe. Tactile Sensing and Control of Robotic Manipulation. *Advanced Robotics*, 8(3):245–261, 1993.

[16] R.D. Howe, N. Popp, P. Akella, I. Kao, and M.R. Cutkosky. Grasping, Manipulation, and Control with Tactile Sensing. In *IEEE International Conference on Robotics and Automation*, 1990.

[17] A.R. Jiménez, A.S. Soembagijo, D. Reynaerts, H. Van Brussel, R. Ceres, and J.L. Pons. Featureless Classification of Tactile Contacts in a Gripper using Neural Networks. *Sensors and Actuators A: Physical*, 62(1-3):488–491, 1997.

[18] Georg Klein and Tom Drummond. Robust Visual Tracking for Non-Instrumented Augmented Reality. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2003.

[19] Georg Klein and David Murray. Full-3D Edge Tacking with a Particle Filter. In *Proc. British Machine Vision Conference (BMVC)*, volume 3, pages 1119–1128, September 2006.

[20] Marek Kopicki, Rustam Stolkin, Sebastian Zurek, Thomas Mörwald, and Jeremy Wyatt. Predicting workpiece motions under pushing manipulations using the principle of minimum energy. In *Proceedings of the RSS workshop on Representations for Object Grasping and Manipulation in Single and Dual Arm Tasks*, 2010.

[21] Marek Kopicki, Rustam Stolkin, Sebastian Zurek, and Jeremy Wyatt. Learning to predict how rigid objects behave under simple manipulation. Technical report, University of Birmingham, 2010.

[22] Marek Kopicki, Sebastian Zurek, Rustam Stolkin Thomas Mörwald, and Jeremy Wyatt. Learning forwards models for the motion of manipulated objects. *submitted to IEEE Transactions on Robotics*, 2011.

[23] Marek Kopicki, Sebastian Zurek, Rustam Stolkin, Thomas Mörwald, and Jeremy Wyatt. Learning to predict how rigid objects behave under simple manipulation. In *Proceeedings of the IEEE International Conference on Robotics and Automation*, pages 5277–5279, 2011.

[24] Danica Kragic, Andrew T. Miller, and Peter K. Allen. Real-time Tracking Meets Online Grasp Planning. In *IEEE International Conference on Robotics and Automation*, pages 2460–2465, 2001.

[25] Danica Kragic, Lars Petersson, and Henrik I Christensen. Visually guided manipulation tasks. *Robotics and Autonomous Systems*, 40(2-3):193–203, 2001.

[26] Junghyun Kwon and F.C. Park. Visual Tracking via Particle Filtering on the Affine Group. In *Int. Conf. on Information and Automation (ICIA)*, pages 997–1002, 2008.

[27] Kevin Lynch. The Mechanics of Fine Manipulation by Pushing. In *IEEE International Conference on Robotics and Automation*, pages 2269–2276, 1992.

[28] M. T. Mason. *Manipulator grasping and pushing operations*. PhD thesis, MIT, 1982.

[29] Lucie Masson, Michel Dhome, and Frederic Jurie. Robust Real Time Tracking of 3D Objects. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 252–255, 2004.

[30] Philipp Michel, Joel Chestnutt, Satoshi Kagami, Koichi Nishiwaki, James Kuffner, and Takeo Kanade. GPU-Accelerated Real-Time 3D Tracking for Humanoid Autonomy. In *Proceedings of the JSME Robotics and Mechatronics Conference (ROBOMEC'08)*, 2008.

[31] A. Miller and P. Allen. Examples of 3D Grasp Quality Computation. In *IEEE International Conference on Robotics and Automation*, pages 1240–1246, 1999.

[32] L. Montesano and M. Lopes. Learning Grasping Affordances from Local Visual Descriptors. In *IEEE International Conference on Development and Learning*, 2009.

[33] A. Morales, M. Prats, P.J. Sanz, and A. P. Pobil. An Experiment in the Use of Manipulation Primitives and Tactile Perception for Reactive Grasping. In *Robotics: Science and Systems, Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

[34] T. Mörwald, M. Kopicki, R. Stolkin, J. Wyatt, S. Zurek, M. Zillich, and M. Vincze. Predicting the Unobservable – Visual 3D Tracking with a Probabilistic Motion Model. In *Proceeedings of the IEEE International Conference on Robotics and Automation*, pages 1849–1855, 2011.

[35] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze. BLORT - The Blocks World Robotic Vision Toolbox. In *Best Practice in 3D Perception and Modeling for Mobile Manipulation (in conjunction with ICRA 2010)*, 2010.

[36] T. Mörwald, M. Zillich, and M. Vincze. Edge Tracking of Textured Objects with a Recursive Particle Filter. In *19th International Conference on Computer Graphics and Vision (Graphicon), Moscow*, pages 96–103, 2009.

[37] Erik Murphy-Chutorian and Mohan M. Trivedi. Particle Filtering with Rendered Models: A Two Pass Approach to Multi-Object 3D Tracking with the GPU. In *CVPR workshop on Computer Vision on GPU's (CVGPU)*, pages 1–8, 2008.

[38] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner. Learning to perceive affordances in a framework of developmental embodied cognition. In *IEEE 6th International Conference on Development and Learning, 2007. ICDL 2007*, pages 110–115, 2007.

[39] M. A. Peshkin and A. C. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Journal on Robotics and Automation*, 4:569–598, 1988.

[40] Lars Petersson, David Austin, and Danica Kragic. High-level Control of a Mobile Manipulator for Door Opening. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2333–2338, 2000.

[41] Anna Petrovskaya, Oussama Khatib, Sebastian Thrun, and Andrew Ng. Bayesian Estimation for Autonomous Object Manipulation based on Tactile Sensors. In *IEEE International Conference on Robotics and Automation*, pages 707–714, 2006.

[42] M. Prats, P.J. Sanz, and A.P del Pobil. Vision-Tactile-Force Integration and Robot Physical Interaction. In *IEEE International Conference on Robotics and Automation*, pages 3975–3980, 2009.

[43] Andreas Richtsfeld, Thomas Mörwald, Michael Zillich, and Markus Vincze. Taking in Shape: Detection and Tracking of Basic 3D Shapes in a Robotics Context. In *Computer Vision Winder Workshop (CVWW)*, pages 91–98, 2010.

[44] B. Ridge, D. Skocaj, and A. Leonardis. Towards learning basic object affordances from object properties. In *Proceedings of the International Conference on Cognitive Systems*, 2008.

[45] J.R. Sánchez, H. Álvarez, and D. Borro. Towards Real Time 3D Tracking and Reconstruction on a GPU using Monte Carlo Simulations. In *9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 185–192, 2010.

[46] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic Grasping of Novel Objects using Vision. *International Journal of Robotics Research*, 27(2):157, 2008.

[47] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard. Object Identification with Tactile Sensors Using Bag-of-Features. In *Proceedings of the International Conference on Intelligent Robot and Systems*, pages 243–248, 2009.

[48] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2004.

[49] B. H. Yoshimi and P. Allen. Closed-loop Visual Grasping and Manipulation. In *IEEE International Conference on Robotics and Automation*, 1996.

[50] Yu Zheng and Wen Qian. Improving Grasp Quality Evaluation. *Robotics and Autonomous Systems*, 57(6-7):665–673, 2009.

Learning to predict how rigid objects behave under simple manipulation

Marek Kopicki[1], Sebastian Zurek[1], Rustam Stolkin[1], Thomas Mörwald[2], Jeremy Wyatt[1]
[1]School of Computer Science, University of Birmingham, UK
[2]Automation and Control Institute, Vienna University of Technology, AT

Abstract— **An important problem in robotic manipulation is the ability to predict how objects behave under manipulative actions. This ability is necessary to allow planning of object manipulations. Physics simulators can be used to do this, but they model many kinds of object interaction poorly. An alternative is to learn a motion model for objects by interacting with them. In this paper we address the problem of learning to predict the interactions of rigid bodies in a probabilistic framework, and demonstrate the results in the domain of robotic push manipulation. A robot arm applies random pushes to various objects and observes the resulting motion with a vision system. The relationship between push actions and object motions is learned, and enables the robot to predict the motions that will result from new pushes. The learning does not make explicit use of physics knowledge, or any pre-coded physical constraints, nor is it even restricted to domains which obey any particular rules of physics. We use regression to learn efficiently how to predict the gross motion of a particular object. We further show how different density functions can encode different kinds of information about the behaviour of interacting objects. By combining these as a product of densities, we show how learned predictors can cope with a degree of generalisation to previously unencountered object shapes, subjected to previously unencountered push directions. Performance is evaluated through a combination of virtual experiments in a physics simulator, and real experiments with a 5-axis arm equipped with a simple, rigid finger.**

I. Introduction

This paper presents algorithms which learn to predict the motion of a rigid object resulting from an robot push. These algorithms do not rely on any encoding of Newtonian mechanics, but can be trained online. Object interactions are learned as distributions. Our system does not know a priori about impenetrability, gravity, or kinematic relations between objects, all being learned from data.

Although work has been done on push manipulation in robots [1], [2], [3], [4] it is restricted to planar sliding motions of what are effectively 2D objects. There is little literature addressing the more complex problem of push manipulations on real 3D bodies, which are free to tip or roll. It is possible to use physics simulators to predict the motion of interacting rigid bodies. However, this approach is reliant on explicit knowledge of the objects, the environment, and key physical parameters which can be difficult to tune. Even then, such predictions may not be possible due to inherent limitations of the physical model employed, for example when modeling friction.

Machine learning approaches have been developed to learn to classify or provide predictions for objects or object classes, e.g. rolling versus non-rolling objects [5], [6], or liftable versus non-liftable objects [7]. These kinds of approach are limited, in that predictions learned may not be generalisable to a new object, pose or push direction, and explicit 6-DOF rigid body motions are not predicted. In contrast, our approach learns to make predictions of explicit 3D rigid body transformations. The probabilistic nature of the learning enables generalisation to novel push directions, object poses, and objects with novel shapes.

This paper extends our previous work [8] in three ways. First, we modify the prediction scheme to make use of local coordinate systems that move with parts of the object. This improves learning and generalisation, since now we predict relative rather than absolute changes in pose. Second, we show how a two expert approach can be extended to include a combination of many experts, which encode new information about how objects interact. This change allows generalisation with respect to both push direction, and object shape. Third, we implement a version of our prediction scheme based on regression, and show how it can efficiently learn the gross motion characteristics of a particular object, although it can struggle with certain kinds of generalisation. Finally we present results from physical experiments in which various real objects were subjected to complex 3D motions, such as tipping and toppling, while pushed by a real robot. The real experiments are additionally supported by an extensive set of simulation experiments.

II. Representations

Consider three reference frames A, B and O in a 3-dimensional Cartesian space (see Figure 1). While frame O is fixed, A and B change in time and are observed at discrete time steps $..., t-1, t, t+1, ...$ every non-zero Δt. A frame X at time step t is denoted by X^t, a rigid body transformation between a frame X and a frame Y is denoted by $T^{X,Y}$.

From classical mechanics we know that in order to predict a state of a body, it is sufficient to know its mass, velocity and a net force applied to the body. We do not assume any knowledge of the mass and applied forces, however the transformations of a body, with attached frame B, over two time steps T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ encode its acceleration - the effect of the applied net force. Therefore, if the net force and the body mass are constant, the transformations T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ provide a complete description of

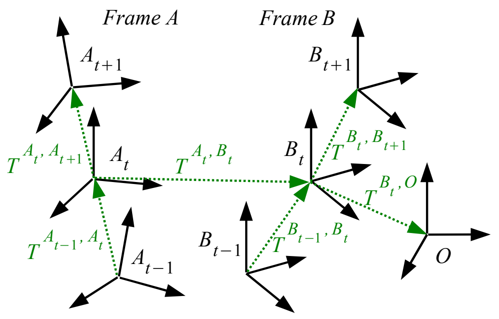


Fig. 1. A system consisting of two interacting bodies with frames A and B in some constant environment with frame O can be described by six rigid body transformations T^{A_t,B_t}, $T^{B_t,O}$, T^{A_{t-1},A_t}, $T^{A_t,A_{t+1}}$, T^{B_{t-1},B_t}, and $T^{B_t,B_{t+1}}$.

the state of a body at time step t in absence of other bodies. A triple of transformations $T^{B_t,O}$, T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ provide a complete description of a state of a body in some fixed frame of reference O which accounts for a constant or stationary environment. Similarly, transformations $T^{A_t,O}$, T^{A_{t-1},A_t} and $T^{A_t,A_{t+1}}$ provide such a description for some other body with frame A.

The state of a system consisting of three bodies with frames A and B in some constant environment with frame O can be described by the six transformations as it is shown in Figure 1, where $T^{A_t,O}$ has been replaced by a relative transformation T^{A_t,B_t}.

The prediction problem can be stated as: given we know or observe the starting states and the motion of the pusher, $T^{A_t,A_{t+1}}$, predict the resulting motion of the object, $T^{B_t,B_{t+1}}$. This is a problem of finding a function:

$$F : T^{A_t,B_t}, T^{B_t,O}, T^{A_{t-1},A_t}, T^{B_{t-1},B_t}, T^{A_t,A_{t+1}} \longrightarrow T^{B_t,B_{t+1}} \tag{1}$$

Function F is capable of describing all possible effects of interactions between rigid bodies A and B, providing their physical properties and applied net forces are constant in time, in the limit of infinitesimally small time steps. Furthermore, it can be approximately learned from observations for some small fixed time interval Δt between time steps.

In this work, we will focus on robotic manipulations that are performed relatively slowly, hence we assume quasi-static conditions, and ignore all frames at time $t-1$. This conveniently reduces the dimensionality of the problem, giving a simplified function, F_{qs}:

$$F_{qs} : T^{A_t,B_t}, T^{B_t,O}, T^{A_t,A_{t+1}} \longrightarrow T^{B_t,B_{t+1}} \tag{2}$$

The behaviours of interacting bodies represented by rigid body transformations as in Figure 1 are independent of their poses with respect to some inertial frame I [9]. Therefore instead of using inertial frame-dependent transformation $T_{in}^{A_t,A_{t+1}}$, one can represent object transformations in the object body frame (see Figure 2). The body frame transformation $T_{body}^{A_t,A_{t+1}}$ is obtained by moving instantaneous frame A, so that at time t it overlaps with inertial frame I. Given some instantaneous object frame A_t at time t, and the transformation $T_{in}^{A_t,A_{t+1}}$, one can obtain transformation $T_{body}^{A_t,A_{t+1}}$ in the body frame (via a similarity transform):

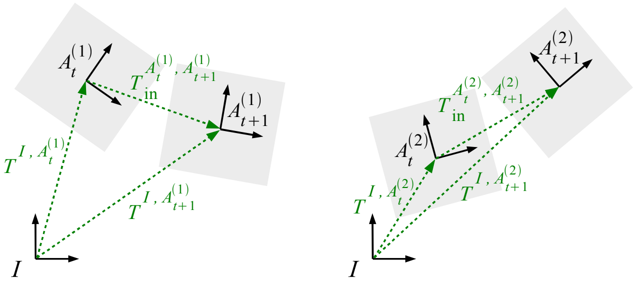


Fig. 2. In the above two scenes a pose change between time step t and $t+1$ as observed in instantaneous object body frame $A^{(1)}$ and the same object in another instantaneous body frame $A^{(2)}$ given inertial frame I are both the same. However because transformations $T^{I,A^{(1)}}$ and $T^{I,A^{(2)}}$ are different, the corresponding transformations in the inertial frame are also different, i.e. $T_{in}^{A_t^{(1)},A_{t+1}^{(1)}} \neq T_{in}^{A_t^{(2)},A_{t+1}^{(2)}}$.

$$T_{body}^{A_t,A_{t+1}} = (T^{I,A_t})^{-1} T_{in}^{A_t,A_{t+1}} T^{I,A_t} \tag{3}$$

where we have made use of the identities $T^{I,A_{t+1}} = T_{in}^{A_t,A_{t+1}} T^{I,A_t} = T^{I,A_t} T_{body}^{A_t,A_{t+1}}$.
Given a transformation in the body frame, instantaneous object frame A_t at t and using Equation (3), transformation $T_{in}^{A_t,A_{t+1}}$ in the inertial frame is given by:

$$T_{in}^{A_t,A_{t+1}} = T^{I,A_t} T_{body}^{A_t,A_{t+1}} (T^{I,A_t})^{-1} \tag{4}$$

In further discussion we will retain subscripts in, but suppress subscripts $body$, and assume that all transformations $T^{X,Y}$ are transformations in the body frame X obtained using a similarity transform $T^{X,Y} \equiv T_{body}^{X,Y} = (T^{I,X})^{-1} T_{in}^{X,Y} T^{I,X}$.

Since the prediction problem is posed as finding a function, we can now apply our function approximator of choice. In this paper we use LWPR [10] - a powerful method applied widely in robotics.

III. Learning Global and Local Experts as Density Estimation

Having now formulated prediction as a function approximation problem, in this section we recast it as a density estimation problem. The motivation for this is that prediction learning using functions F or F_{qs} is limited with respect to changes in shape and type of manipulation.

Consider a 2D projection at time t of a robotic finger with global frame A_t, an object with global frame B_t, and the constant global frame O (Figure 3). We can identify local frames A_t^l and B_t^l, rigidly attached to small local planar surface patches at the contact point, or the points of closest proximity on the object and finger. We define the global information to be the information about changes of the pose of the whole object, whereas the local information is specified by changes in the local frames A_t^l and B_t^l.

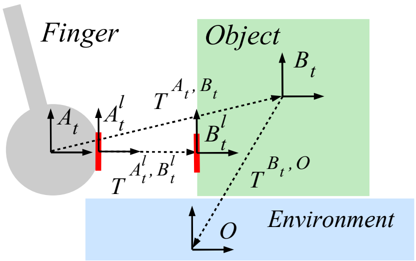


Fig. 3. 2D projection at time t of a robotic finger with global frame A_t, an object with global frame B_t, and a ground plane with constant global frame O. Local frames A_t^l and B_t^l describe the local shape of the finger and an object at their point of closest proximity.

In order to combine both global and local information, one can incorporate contact information represented by transformations $T^{A_t^l,A_{t+1}^l}$ and $T^{A_t^l,B_t^l}$ directly into the domain of function F_{qs}. This, however, would significantly increase the dimensionality of the function's domain. Instead, we recast the mapping F_{qs} as a conditional probability $P_{qs}(T^{B_t,B_{t+1}}|\cdot)$, i.e. a probability density over rigid body transformations of the object [8]. This reformulation allows us to combine the global and local information as a *product of densities* to approximate P_{qs}, so that (schematically, for some normalisation constant N)

$$P_{qs} \approx N \; P_{global} \; P_{local} \tag{5}$$

where

$$P_{global} \equiv P_{global}(T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}}, T^{A_t,B_t}, T^{B_t,O}) \tag{6}$$

$$P_{local} \equiv P_{local}(T^{B_t^l,B_{t+1}^l}|T^{A_t^l,A_{t+1}^l}, T^{A_t^l,B_t^l}) \tag{7}$$

denote the global and local density functions or "experts" [8]. The densities P_{global} and P_{local} factorise the conditioning variables of P_{qs}, and hence manage the complexity of incorporating more information into the predictor.

The above global and local densities encode information about which candidate rigid body transformations are more or less feasible for each frame of reference respectively. However, once we form the product of these two densities, only transformations which are feasible in both frames will have high probability in the resulting combined distribution.

The rationale for introducing global and local experts, instead of using a straightforward function approximation, can be explained by considering a backward-push experiment as shown in Figure 4.

The configuration of finger and object during a backward push is very different to those present in a training set consisting only of forward pushes. A predictor comprised of just a global expert will fail to generalize to a new push direction that differs markedly from any observed in the training set for the expert. However, by also using the local expert P_{local}, the predictor can learn that the finger does not penetrate the object after contact. Any candidate motion preferred by the global expert will be 'vetoed' by the local expert if impenetrability is violated. Nevertheless, there are other constraints on the object motion, such as the ground plane, which are not encoded by the local expert. To model these other facts about possible object motion requires the use of additional experts as described in the next section.

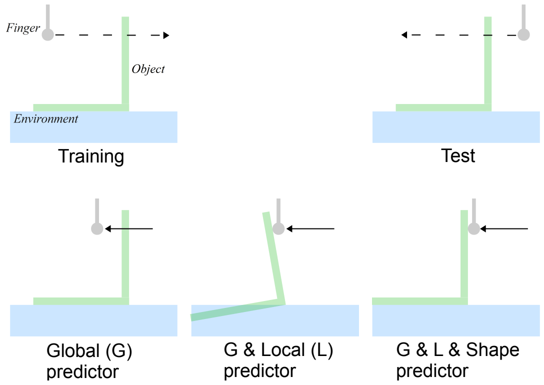


Fig. 4. Schematic diagram (2D projection of 3D scene) in which an object (of L-shaped cross-section) on a supporting surface is pushed by a robotic finger. Various predictors are trained solely on forward pushes (top left), but tested on backwards pushes (top right). The top panels show the push trajectory for the training and test phases, whereas the bottom panels show the outputs from three types of predictor in the test phase. A predictor comprised of just a global expert will fail to generalize, and will predict that the object does not move as the finger passes through it (bottom left). Adding a local expert will stop the finger penetrating the object, but does not guarantee that the predicted object motion will respect other impenetrability constraints (bottom middle). Finally, using an additional 'local shape' expert attached to the base of the object, a physically plausible motion is obtained (bottom right).

Returning to the formal development, we now consider the relations between transformations expressed in the body frame of the local patches and corresponding transformations in the inertial frames. For coordinate frames as shown shown in Figure 3, from object rigidity and using Equation (3) we have:

$$T^{A_t^l,A_{t+1}^l} = (T^{I,A_t^l})^{-1} T_{in}^{A_t,A_{t+1}} T^{I,A_t^l} \tag{8a}$$

$$T^{B_t^l,B_{t+1}^l} = (T^{I,B_t^l})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t^l} \tag{8b}$$

where I is the inertial frame. $T^{A_t^l,B_t^l}$ can be determined directly from the shape frame:

$$T^{A_t^l,B_t^l} = (T^{I,A_t^l})^{-1} T_{in}^{A_t^l,B_t^l} T^{I,A_t^l} \tag{9}$$

For the finger-object scenario, a prediction problem can then be defined as finding that transformation $\widetilde{T}_{in}^{B_t,B_{t+1}}$ in the inertial frame which maximises the product of the two conditional densities (experts) (6) and (7):

$$\widetilde{T}_{in}^{B_t,B_{t+1}} = \underset{T_{in}^{B_t,B_{t+1}}}{\operatorname{argmax}} \left\{ P_{global} \; P_{local} \right\} \tag{10}$$

where the similarity transforms (3) (in frame B_t) and (8b) must be used to evaluate P_{global} and P_{local} for a given $T_{in}^{B_t,B_{t+1}}$.

Starting with some initial state of the finger T^{A_0} and object T^{B_0}, and knowing the trajectory of the finger $A_1, \ldots A_T$ over T time steps, one can predict a whole trajectory of the object $B_1, \ldots B_T$, by iterating the prediction obtained from

Equation (10). That is, the output of the prediction at time t is used as input to the prediction for the next time step.

IV. Incorporating information from additional experts

In addition to learning how an object moves in response to a push, it is desirable to incorporate learned information about the inherent tendencies of parts of an object to move in various directions with respect to the environment or other objects, regardless of whether the object is being pushed or not. This additional information may help when predicting the motion of a previously unseen object, or the response to a novel push direction (Figure 4), because it provides some prior knowledge about which kinds of motions are possible and which are not.

We can incorporate this additional information by attaching an arbitrary number of additional coordinate frames B^{sn_t} to various parts of the object (Figure 5).

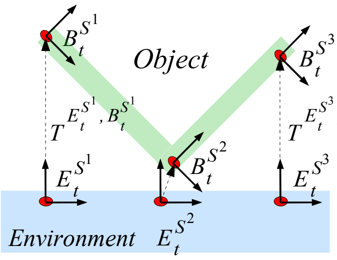


Fig. 5. Co-ordinate frames can be attached to an arbitrary number of local shapes, and local experts can be learned for each of these frames, predicting a distribution of how the frame may move next, given where it is at the present time step.

We then learn densities, also known as local shape experts, for the future motions of each of these frames. To obtain the results presented in this paper, the number and location of local shape experts on each of the different objects were determined by hand.

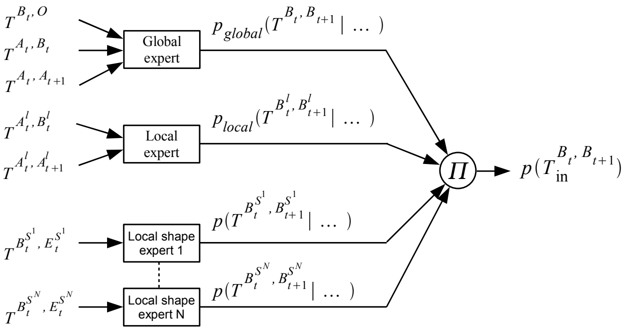
The local shape densities are conditioned only on their relative pose $T^{E^{S^k_t}, B^{S^k_t}}$ with respect to a corresponding pose $E^{S^k_t}$ of a patch on a ground plane at the present time step, ignoring any information about the motions of the pushing finger. For the k-th such frame, we estimate *the local shape conditional density*:

$$P_{shape,k} \equiv P_{shape}(T^{B^{S^k_t}, B^{S^k_{t+1}}} | T^{E^{S^k_t}, B^{S^k_t}}) \tag{11}$$

which represents the probability density over possible rigid body transformations in the body frame of the k-th local contact. Analogous to Equation (10), the subsequent motion of the object in the inertial frame can be predicted as:

$$\widetilde{T}_{in}^{B_t, B_{t+1}} = \underset{T_{in}^{B_t, B_{t+1}}}{\operatorname{argmax}} \left\{ P_{global}\, P_{local} \prod_{k=1\ldots N} P_{shape,k} \right\} \tag{12}$$

where N is the number of local shape experts (Figure 6).



$T^{B_t, O}$
T^{A_t, B_t}
$T^{A_t, A_{t+1}}$
Global expert
$p_{global}(T^{B_t, B_{t+1}} | \ldots)$
$T^{A^l_t, B^l_t}$
$T^{A^l_t, A^l_{t+1}}$
Local expert
$p_{local}(T^{B^l_t, B^l_{t+1}} | \ldots)$
Π
$p(T_{\text{in}}^{B_t, B_{t+1}})$
$T^{B_t^{S^1}, E_t^{S^1}}$
Local shape expert 1
$p(T^{B_t^{S^1}, B_{t+1}^{S^1}} | \ldots)$
$T^{B_t^{S^N}, E_t^{S^N}}$
Local shape expert N
$p(T^{B_t^{S^N}, B_{t+1}^{S^N}} | \ldots)$

Fig. 6. Inputs and outputs of learned prediction system. The two-expert approach can be extended to include opinions from multiple local shape experts represented by coordinate frames S^N.

V. Implementation

We have now presented two formulations of the prediction learning problem: 1) as function approximation, and 2) as density estimation. We have suggested that there may be an advantage to solving the density problem by applying the heuristic of a product of experts (densities).

Regression method. We used LWPR [10] to estimate the mapping described by Equation (2). The regression scheme was implemented using the LWPR software library [11].

Single expert and multiple expert methods. A variant of Kernel Density Estimation is used to approximate conditional densities in (12) (for details see [8]). The single expert method employed only a global expert (6). The density product (12) is maximised using the differential evolution optimisation algorithm [12], which has a run time proportional to the product of the population size and the number of generations used in the algorithm. The run time also scales linearly with the number of experts, the number of kernels and the number of parameters used to encode the rigid body transform.

Rigid body transformations used in both learners were parametrised by 6 numbers: Euler angles and a displacement.

VI. Experimental study

We have tested our prediction algorithms in a number of experiments (see section C), in which a real robot arm applies pushes to various real objects. The arm has accuracy of ±1mm in the region of the contacts in the reported experiments, and the predictors are trained on poses captured by a particle filter based tracker, which has pose errors of the order of ±2mm frames for most frames, with up to ±5mm in 5% of frames for some videos where the polyflap object is beginning to tip over. These tracking errors are significantly smaller than the average prediction errors generated by any of the predictors (± 20 to 80mm) as well as the differences between those average prediction errors (± 6 to 50mm).

Section D presents the results of simulation experiments, which are designed to test the ability of learned predictors to generalise in various different ways. The simulation environment usefully provides us with perfect ground-truth data against which to evaluate predictions, and also enables a very large number of experiments with many different values of key parameters (e.g. shape of pushed objects). Replication of the experiments in Section D on the real robot is planned future work.

Section C shows that the virtual environment (using NVIDIA PhysX) does not replicate the physical properties of the real world perfectly. We hand tuned the parameters of the physics engine to best fit the world, and in principle this could also be done automatically. However, we have found that even when optimised, the parameters neither correspond to their true values, nor do they generalise well. However, regardless of how well they correspond to the real world, the simulations still provide a self-consistent experimental environment within which to compare the accuracy of predictors that have been trained within that environment.

A. Setup

Multiple experimental trials were performed, in which a robotic arm equipped with a finger performs a random pushing movement towards an object (Figure 7). In each experiment data samples are stored over a series of such random trials. Each trial lasts exactly 10 seconds, while data samples are stored every $1/15$th of a second.

For real experiments, we use a 5-axis Katana robotic manipulator [13] equipped with a single rigid finger, and the motion of pushed objects is captured using a single camera and a visual tracking algorithm [14]. Simulation experiments are carried out using the NVIDIA PhysX physics engine [15].

Local shape experts in the multiple expert method were fixed by hand to a L-shaped object (referred to as "polyflap") as it is shown in Figure 5). In the case of a box-shaped object (Experiments 3 and 5), there were 4 local shape experts fixed to the edges of a box.

The bandwidth of all distributions used in the multiple experts method as well as parameters of the LWPR regression method were tuned once by hand and kept constant throughout all the experiments.

B. Performance measure

In all experiments, we take the output of the tracked 6D pose of a real object to be ground-truth, and compare it against predictions which were previously forecast by the learned prediction system. The vision system does not provide perfect ground-truth, yielding typical errors of around ±2mm during successful tracking, or arbitrarily large errors when the track is occasionally lost. However, comparing predictions to the outputs of the tracker still provides some useful information about discrepancies in the predictor, although clearly the performance of the predictors is limited by the accuracy of the data on which they are trained. Prediction performance is evaluated as follows.

At any particular time step, t, a large number, N, of randomly chosen points $p_n^{1,t}$, where $n = 1 \ldots N$, are rigidly attached to an object at the ground-truth pose, and the corresponding points $p_n^{2,t}$ to an object at the predicted pose. At time step t, an average error E_t can now be defined as the mean of displacements between points on the object at the predicted pose and points on the object at the ground-truth pose:

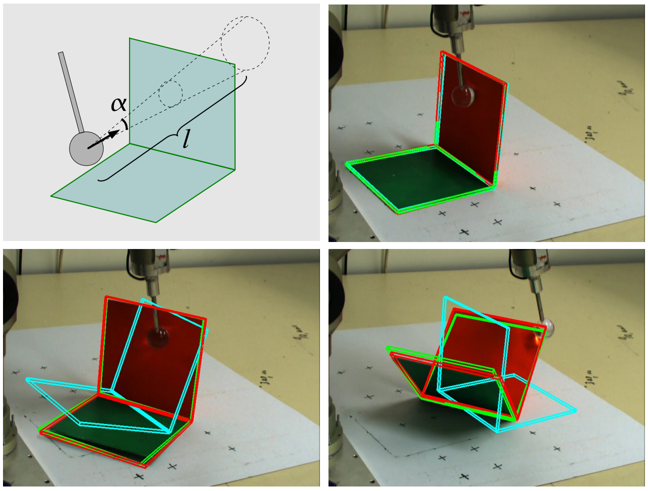


Fig. 7. A 5-DOF robotic arm equipped with a finger performs a random straight-line pushing movement of a variable length l=25±5 cm within a cone with angle α=20 deg towards an object (top left). The movement begins at a random location so that every small region on the upper part of an object is equally likely to be pushed. The object behaviour can be complex and varies depending on the finger trajectory and its pose relative to the object. In the image sequence shown above, the object begins to rotate anti-clockwise (top right - bottom left) before tilting (bottom right). The red wire-frame shows the output from the vision tracking system. The green wire-frame indicates the object pose predicted by the multiple-expert learning method, while the blue wire-frame is generated by the PhysX simulator. Although the PhysX predictions are qualitatively plausible, it was virtually impossible to tune the simulator so that its predictions match reality for all training data. Note that the entire motion sequence is predicted before the physical push is initiated, without any correction from visual feedback during the push execution.

$$E_t = \frac{1}{N} \sum_{n=1 \ldots N} |p_n^{2,t} - p_n^{1,t}| \tag{13}$$

Note that for each robotic push action, we predict approximately 150 consecutive steps into the future, with no recursive filtering or corrector steps, hence it is expected that errors will grow with range from the initial object pose. We therefore find it more meaningful to normalise all errors with respect to an "average range", R_t, of the object from its starting position, defined as:

$$R_t = \frac{1}{N} \sum_{n=1 \ldots N} |p_n^{1,t} - p_n^{1,0}| \tag{14}$$

For a test data set, consisting of K robotic pushes, each of which breaks down into many consecutive predictions over T time steps, we can now define average error and normalised average error:

$$E_{av} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{t=1}^{T} E_t, \quad E_{av}^{norm} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{T} \sum_{t=1}^{T} \frac{E_t}{R_t} \tag{15}$$

For each set of test data, we also report final error and normalised final error, which represent the typical discrepancy between prediction and ground truth that has accumulated by the end of each full robotic push:

$$E_f = \frac{1}{K}\sum_{k=1}^{K} |p_n^{2,T} - p_n^{1,T}|, \quad E_f^{norm} = \frac{1}{K}\sum_{k=1}^{K} \frac{|p_n^{2,T} - p_n^{1,T}|}{R_T} \tag{16}$$

Note that both normalised errors have no units.

We performed 10-fold cross-validation where at the beginning of each experiment all the trials are randomly partitioned into 10 subsets. Prediction was then subsequently performed (10 times) on each single subset, while learning (only for learned approaches) was always performed on the remaining 9 subsets of these trials. All the results were then averaged to produce a single estimation.

C. Experiments with a real robot

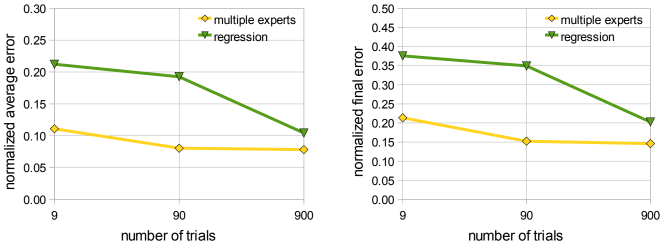


Fig. 8. Experiment 1 with a real robot and a polyflap object. Decrease in average (left) and final (right) prediction errors with increasing number of learning trials, for two different prediction methods.

Experiment 1: comparison of learning methods for a real robot pushing a polyflap object. We have trained the system on 9, 90 and 900 pushes of a polyflap object with a real robotic finger (Figure 7). We evaluated the performance of the multiple expert and regression methods. Figure 8 shows that the average and final prediction error decreases with increased number of trials used in learning for both tested prediction methods. The multiple expert method performed reasonably well, even when trained on as little as 9 example pushes. The method performed particularly well with 90 learning trials, as local experts successfully prevented the predictor from violating impenetrability constraints that were frequently violated by the regression method. However, the performance of the multiple expert method did not significantly improve with 900 learning trials. One of the reasons for this is that the visual tracking system is far from perfect. The tracking often contains significant errors, and the quality of tracking is not pose-independent. For example, cases of tipping and toppling movements are particularly difficult to track, so that the prediction system does not always have sufficiently accurate training data to precisely learn all possible motions.

Additionally we obtained predictions using the NVIDIA PhysX physics simulator, with parameters hand-tuned to match the real system. Figure 9 presents a comparison of the physics simulation and the learned predictors (trained on 900 trials). Clearly, the physics simulator is unable to match predictors trained in a real experiment, even though the real training data contains significant errors due to occasional failures and inaccuracies of the vision system. In particular, the physics simulator has difficulty modelling the frictional interactions of the real world, and often is unable to accurately simulate a rotational movement of the object.

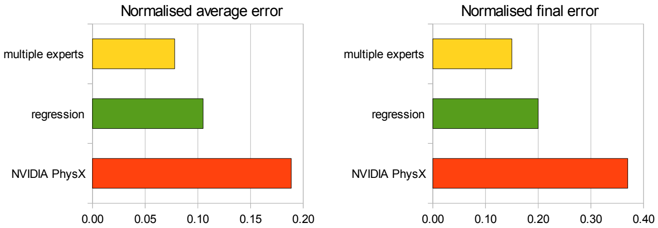


Fig. 9. Experiment 1. Physics simulation is unable to match the performance of learned predictors which have been trained in real experiments.

Experiment 2: comparison of learning methods for a real robot pushing a small box. We have trained the system on 9, 90 and 450 pushes of a small box object with a real robotic finger. Figures 11 and 12 show examples of the multiple expert method making accurate predictions of the box motion when it topples and when it rotates under manipulative pushes. As with Experiment 1, the learning converges within a few hundred example pushes. The multiple expert performed reasonably well, even when trained on as little as 9 example pushes.

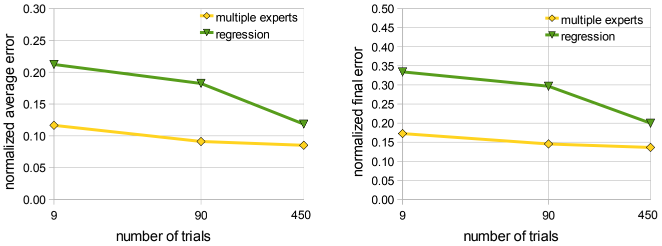


Fig. 10. Experiment 2 with a real robot pushing a small box object. Decrease in average (left) and final (right) prediction errors with increasing number of learning trials, for two different prediction methods.

D. Experiments in a virtual environment

Experiment 3: extrapolative generalisation of pushing directions. In this experiment, a virtual robotic arm applied random orthogonal and oblique pushes to the outside of a polyflap which were then used in training. In contrast, the system was tasked to make predictions for previously unencountered pushes – those applied to the inside surface of the polyflap (thus pushing in the opposite direction to the training pushes). We consider this to be a test of “extrapolative” action generalisation, in that the push directions used

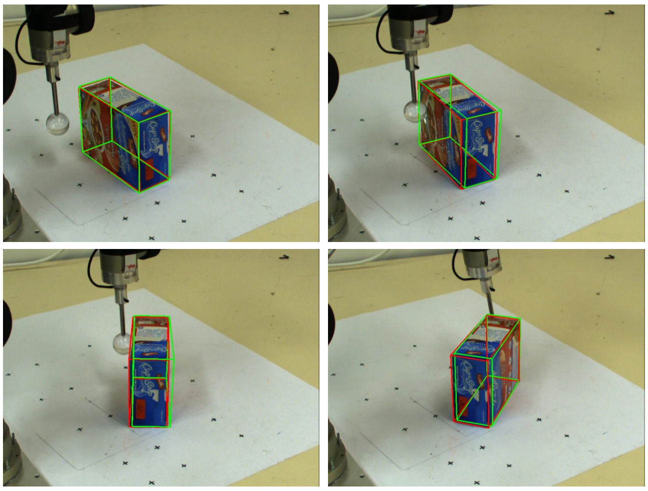


Fig. 11. Experiment 2: accurate predictions of the motion of a small box, as it translates and rotates under a manipulative push from a real robot. The green wire-frame indicates the predicted object pose; the red wire-frame shows the tracked pose from the vision system.

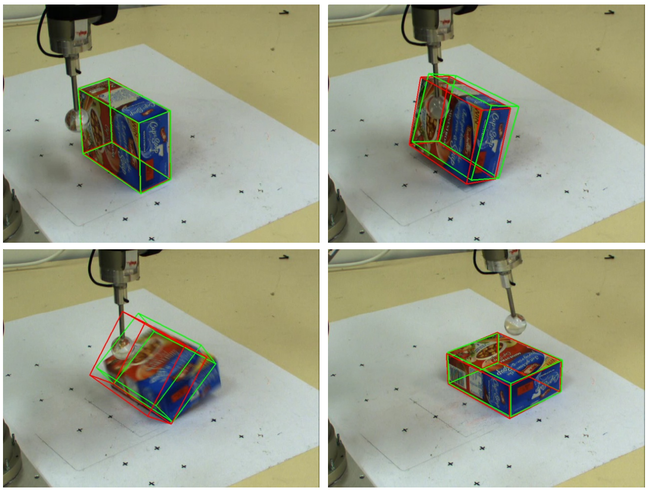


Fig. 12. Experiment 2: accurate predictions of the motion of a small box, as it topples over, under a manipulative push from a real robot. The green wire-frame indicates the predicted object pose; the red wire-frame shows the tracked pose from the vision system.

in testing are all qualitatively different from those used in training – the test push directions do not lie in the same region of data covered by the training examples. The regression and single expert methods failed to predict the polyflap behaviour, and gave physically implausible predictions in which the fingertip penetrated the polyflap (Figure 13). In contrast, the multiple expert method gave a relatively accurate prediction, in which even inaccurate portions of the object trajectory were still physically plausible, and did not violate basic physical constraints on object behaviour such as impenetrability (Figure 13). Note that the motion model is entirely learned – there was no pre-programming of Newtonian laws of motion, gravity, the ground plane, or impenetrability constraints.

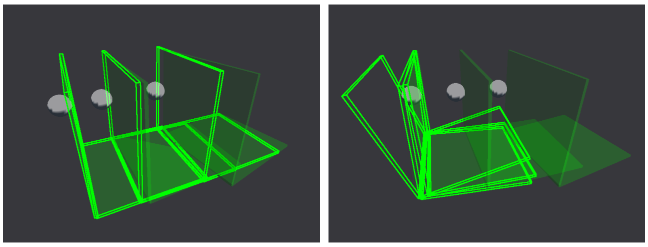


Fig. 13. Experiment 3. Simulation experiment in which predictors are trained only on pushes applied to the inside of the polyflap (moving from right to left in the figure), but are then tested on pushes applied to the outside of the polyflap (i.e. from left to right). The multiple expert method (left panel) predicts a rightwards movement, that comes close to the true motion, does not violate impenetrability, and is physically plausible. In contrast, the regression method (right panel) erroneously predicts that the fingertip (shown as a ball) will pass right through the polyflap. The ground-truth and the predicted poses are shown as solid and wire-frame shapes respectively.

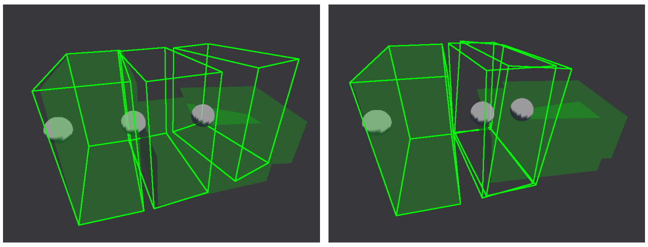


Fig. 14. Experiment 4. Simulation experiment in which predictors have been trained on a polyflap, but tasked with making predictions for a box. The multiple expert method (left panel) predicts a motion which is erroneous (i.e. fails to predict toppling in this case), but is in the correct direction, is physically plausible, and does not violate impenetrability constraints. In contrast, the regression method (right panel) violates impenetrability constraints, as does the single expert method (not shown). The ground-truth and the predicted poses are shown as solid and wire-frame shapes respectively.

Experiment 4: extrapolative generalisation to novel shapes. In this experiment, the predictors were trained on a polyflap, but were then tasked with predicting the motion of a box - a new shape which had never been encountered in training. This is a test of “extrapolative” shape generalisation. The multiple expert method correctly predicts the direction of motion of the box, and makes a physically plausible prediction (but fails to predict that the box should topple over) (Figure 14). In contrast, the regression and single expert methods constantly violate physics, predicting that the fingertip will penetrate right through the box.

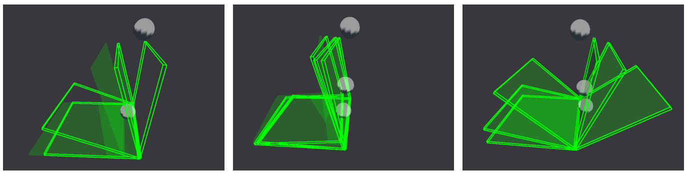


Fig. 15. Experiment 5 reveals limitations of the regression and single expert methods, which fail to predict the motion of a polyflap when subjected to a downward push (left panel). The multiple expert method can cope well with this kind of shape variation (middle and right panels). The ground-truth and the predicted poses are shown as solid and wire-frame shapes respectively.

Experiment 5: interpolative generalisation to novel shapes. This is a virtual experiment, in which all training and

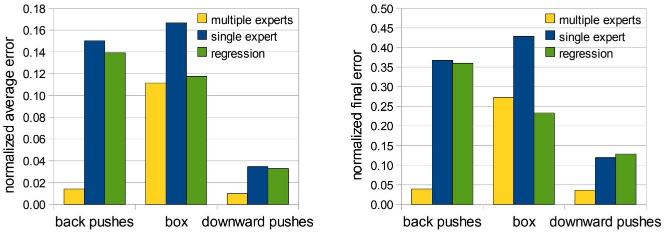


Fig. 16. Experiments 3, 4 and 5. Action generalisation errors for back pushes (Experiment 3), shape generalisation errors for a box (Experiment 4), and downward pushes (Experiment 5).

Exp	**Trials**	**Predictor**	E_{av} [m]	E_{av}^{norm}	E_f [m]	${E_f}^{norm}$
1	900	Multi exp.	**0.021**	**0.078**	**0.036**	**0.146**
	900	Regression	0.027	0.104	0.050	0.206
	n/a	PhysX	0.044	0.189	0.083	0.372
2	450	Multi exp.	**0.023**	**0.085**	**0.037**	**0.136**
	450	Regression	0.032	0.118	0.056	0.200
3	900	Multi exp.	**0.005**	**0.014**	**0.015**	**0.039**
	900	Single exp.	0.054	0.150	0.143	0.367
	900	Regression	0.051	0.139	0.141	0.360
4	900	Multi exp.	**0.042**	**0.111**	0.103	0.272
	900	Single exp.	0.064	0.167	0.169	0.429
	900	Regression	0.045	0.118	**0.093**	**0.233**
5	900	Multi exp.	**0.002**	**0.009**	**0.008**	**0.036**
	900	Single exp.	0.007	0.035	0.023	0.119
	900	Regression	0.007	0.033	0.026	0.129

TABLE I

COMPARATIVE PERFORMANCE OF TESTED PREDICTORS. E_{av} AND E_f ARE MEASURED IN METRES. THE OTHER MEASURES ARE UNITLESS AS EXPLAINED ABOVE.

testing data involve polyflaps constructed from two square flanges. Random shape variation consists in varying the angle at which the two square flanges are connected along a common edge. This shape variation is very significant - dramatically changing the finger-object contact relations. For example, depending on small changes in the angle of the flanges, the same push from above might cause the entire object to move either leftwards or rightwards (Figure 15). The experiment reveals limitations of the regression and single expert methods. Since these methods do not encode information about the contact variability, they do not generalise well in situations where small changes in shape can cause significant and qualitative changes in the resulting motion, even when the robotic push is the same. In contrast, the product of experts technique copes much better with this kind of shape generalisation. We consider this a form of "interpolative" generalisation task, in that the test and training shapes are qualitatively similar and the range of test shapes can be considered to be spanned by the range of training examples. The results are presented in Figure 16.

VII. CONCLUSIONS

This paper has presented several methods by which a robot can learn to predict the motions of a rigid object that will result from manipulative pushing actions. We have shown how regression can be used to efficiently learn the overall "global" motion of a body. We have further shown how multi-modal distributions of local parts of the motion can be learned by Kernel Density Estimation, and how many of these "local" experts can be combined as a product of densities, significantly extending the capabilities of the system with respect to generalization.

This is the first work of which we are aware, in which explicit predictions of 3D object motions under push manipulation are enabled without hard coding of Newtonian physics and physical constraints, but rather by learning based on simple proprioceptive sensing and visual observations of manipulated bodies. The learning approach significantly outperforms approaches based on physics simulators which often model real world interactions poorly, and which rely on physical parameters which may not be known. Furthermore, the proposed multiple expert approach provides a degree of generalisation with respect to changes in shape and applied actions.

VIII. ACKNOWLEDGMENTS

We gratefully acknowledge the support of EU-FP7-IST grants Nos 248273 (GeRT) and 215181 (CogX).

REFERENCES

[1] M. T. Mason, *Manipulator grasping and pushing operations*. PhD thesis, MIT, 1982.

[2] K. Lynch, "The mechanics of fine manipulation by pushing," in *IEEE Int. Conf. on Robotics and Automation*, pp. 2269–2276, 1992.

[3] M. A. Peshkin and A. C. Sanderson, "The motion of a pushed, sliding workpiece," *IEEE Journal on Robotics and Automation*, vol. 4, pp. 569–598, 1988.

[4] D. J. Cappelleri, J. Fink, B. Mukundakrishnan, V. Kumar, and J. C. Trinkle, "Designing open-loop plans for planar micro-manipulation," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 637–642, 2006.

[5] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action-initial steps towards artificial cognition," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, vol. 3, 2003.

[6] B. Ridge, D. Skocaj, and A. Leonardis, "Towards learning basic object affordances from object properties," in *Proceedings of the 2008 International Conference on Cognitive Systems*, 2008.

[7] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner, "Learning to perceive affordances in a framework of developmental embodied cognition," in *IEEE 6th International Conference on Development and Learning, 2007. ICDL 2007*, pp. 110–115, 2007.

[8] M. Kopicki, J. Wyatt, and R. Stolkin, "Prediction learning in robotic pushing manipulation," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, 2009.

[9] M. Kopicki, *Prediction learning in robotic manipulation*. PhD thesis, University of Birmingham, 2010.

[10] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, 2005.

[11] S. Klanke, S. Vijayakumar, and S. Schaal, "A library for locally weighted projection regression," *The Journal of Machine Learning Research*, vol. 9, pp. 623–626, 2008.

[12] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[13] Neuronics AG, "Katana user manual and technical description." http://www.neuronics.ch, 2004.

[14] T. Mörwald, M. Zillich, and M. Vincze, "Edge tracking of textured objects with a recursive particle filter," in *Proceedings of the Graphicon 2009*, (Moscow, Russia), 2009.

[15] NVIDIA PhysX, "Physics simulation for developers." http://developer.nvidia.com/object/physx.html, 2009.

Predicting the Unobservable
Visual 3D Tracking with a Probabilistic Motion Model

Thomas Mörwald[2], Marek Kopicki[1], Rustam Stolkin[1], Jeremy Wyatt[1], Sebastian Zurek[1]
Michael Zillich[2] and Markus Vincze[2]

[1]School of Computer Science, University of Birmingham, UK
[2]Automation and Control Institute, Vienna University of Technology, AT

Abstract— **Visual tracking of an object can provide a powerful source of feedback information during complex robotic manipulation operations, especially those in which there may be uncertainty about which new object pose may result from a planned manipulative action. At the same time, robotic manipulation can provide a challenging environment for visual tracking, with occlusions of the object by other objects or by the robot itself, and sudden changes in object pose that may be accompanied by motion blur. Recursive filtering techniques use motion models for predictor-corrector tracking, but the simple models typically used often fail to adequately predict the complex motions of manipulated objects. We show how statistical machine learning techniques can be used to train sophisticated motion predictors, which incorporate additional information by being conditioned on the planned manipulative action being executed. We then show how these learned predictors can be used to propagate the particles of a particle filter from one predictor-corrector step to the next, enabling a visual tracking algorithm to maintain plausible hypotheses about the location of an object, even during severe occlusion and other difficult conditions. We demonstrate the approach in the context of robotic push manipulation, where a 5-axis robot arm equipped with a rigid finger applies a series of pushes to an object, while it is tracked by a vision algorithm using a single camera.**

I. Introduction

This paper describes a novel approach to visual tracking of an object undergoing pushing manipulation operations, in which a robot arm, equipped with a single rigid finger, applies a series of pokes or pushes to the object, causing it to move from one pose to another. It may seem somewhat esoteric to focus on pushing, however there are good reasons to do so. Pushing is a very fundamental form of manipulation. More complex activities, such as dexterous in-hand manipulation with multiple fingers, can be viewed as combinations of many simultaneous single-finger actions. More practically, even industrial pick and place operations with a simple two jawed gripper often result in a pushing phase, where uncertainties in both object and robot pose lead to one jaw contacting the object before the other. This can even lead to gross grasp failures when the object topples over under pushing from the first jaw, before the second jaw makes contact. Therefore it is important to solve the problems of robotic pushing, and controlled pushing will typically be reliant on tracking the object pose with a vision system.

II. Related work

There is a limited body of literature describing vision algorithms tailored specifically for robotic manipulation tasks. For example, Drummond and Cipolla [1] incorporate knowledge of kinematic constraints into tracking, to better track articulated chains of rigid bodies, with a view to tracking robotic arms for visual servoing. However, it is more usual for researchers to simply take a generic tracking algorithm and incorporate it with an existing manipulation planning system, e.g. [2]. Typically the vision algorithms are drawn from the model based tracking literature, for example [3–5], which predominantly track by choosing candidate poses of the tracked body, whose projected wire-frame edges best match edges extracted from images. More recently, the ability to make use of advanced graphics cards for high speed projective calculations, means that such techniques can be applied to tracking with robust particle filters, e.g. [6–8].

Particle filters rely on motion models, to propagate particles from one predictor-corrector step to the next. In practice, little may be known about the motion of the tracked object, and so predominantly these motion models must be very blunt instruments. It is typical to simply apply Gaussian noise to particles to propagate them, assuming no real understanding of how the object might move at the next time step. However, if the tracked object is subject to robotic manipulation, we should be able to make use of our knowledge of the planned manipulative action, to make a much more informed prediction of the next phase of the objects motion. In the case of an object which is rigidly held in the jaws of a hand or gripper, the motion prediction problem becomes trivial since the object is exactly constrained to follow the motion of the manipulating arm. However, in pushing manipulation, the motion of an object which will result from an applied single-finger push or poke is much more uncertain.

Early approaches to predicting the effects of robotic pushes on object motion, [9–13], attempted analytical solutions of physical constraints. These approaches did not progress beyond anything more complex than the simple 2D case, with flat polygonal objects, constrained to slide on a planar surface. More recently, Cappelleri [14] used physics simulation software to plan manipulative pushes, but again this was limited to a 2D problem, with a small, flat rectangular object which was constrained to slide while floating on a film of oil to simplify frictional interactions. We know of little in the way of literature which specifically addresses the prediction problem in robotic push manipulations of real 3D objects, which are subject to complex 6-dof motions such as tipping and toppling over. It is possible to use physics simulators to predict the motions of interacting rigid bodies, however this approach is reliant on explicit knowledge of the objects, the environment and key physical parameters which can be surprisingly difficult to effectively tune in practice, [15]. Furthermore, once a physics simulator has been set up for a particular scenario, it is not generalizable to new objects or novel situations.

In contrast, our recent work, [16] proposes a system which can learn to predict the explicit 3D rigid body transformations that will result when an object in an arbitrary orientation is subjected to an arbitrary push. The system does not make use of any physics simulation, or any hard coding of Newtonian physics equations or physical constraints. Instead, a statistical relationship between applied pushes and resulting object motions is trained, by simply having the robot apply a series of random pushes to the object, proprioceptively recording the finger trajectories, and observing the resulting object motions with a vision system.

In this paper, we provide an overview of the "learning-to-predict" architecture, and then show how it can be conveniently incorporated into a particle filter-based vision algorithm to propagate particles from one frame to the next. We demonstrate the effectiveness of the technique, for tracking pushed objects past large occlusions and other difficult circumstances, where attempting vision without adequate prediction would fail.

III. Overview

The paper proceeds as follows. Section IV provides an overview of our system for learning to predict the outcomes of manipulative pushes. We describe how the motions of rigid bodies are represented by coordinate frames and transformations. We show how objects and their motions can be decomposed and how a variety of probabilistic *experts* can be trained to predict various aspects of these motions. We show how to combine the opinions of these experts as a product of densities, which is capable of significant generalization to new objects with different shapes and different push directions which have not been encountered during training. In Section V we describe our algorithm [17] for visual tracking of 3D objects using edges, colour and texture features. We then show how tracking can be improved by incorporating a well trained predictor as described in Section IV. Section VI presents results of this work, providing examples of how the enhanced tracker copes with difficult situations such as occlusion and motion blur. Section VII summarizes the results, and discusses ongoing and future work.

IV. Prediction

This section is just a brief overview of the work in [16, 18] to show how rigid body movement can be described in a probabilistic form.

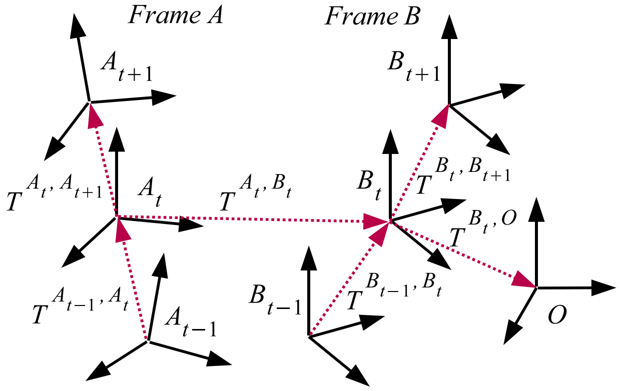


Fig. 1. A system consisting of three interacting bodies with frames A and B and some constant environment with Frame O.

A system consisting of three interacting rigid bodies can be described by coordinate frames A, B and O and by six transformations between the bodies and different time steps ($t-1$, t and $t+1$), with respect to a constant environment O as shown in Figure 1. A and B change in time and are observed at discrete time steps $\ldots, t-1, t, t+1, \ldots$ every non-zero Δt. As stated in [16] a triple of transformations $T^{A_t,O}$, T^{A_{t-1},A_t} and $T^{A_t,A_{t+1}}$ provide a complete description of a state of a rigid body A in terms of classical mechanics. Of course the same is true for some body B. The prediction problem can be stated as: given we know or observe the starting states and the motion of the pusher, $T^{A_t,A_{t+1}}$, predict the resulting motion of the object, $T^{B_t,B_{t+1}}$. This is a problem of finding a function:

$$f : T^{A_t,B_t}, T^{B_t,O}, T^{A_{t-1},A_t}, T^{B_{t-1},B_t}, T^{A_t,A_{t+1}} \to T^{B_t,B_{t+1}} \tag{1}$$

In many robotic applications manipulations are slow, so we can assume quasi-static conditions and it is often possible to ignore all frames at time $t-1$. This conveniently reduces the dimensionality of the problem, giving:

$$f : T^{A_t,B_t}, T^{B_t,O}, T^{A_t,A_{t+1}} \to T^{B_t,B_{t+1}} \tag{2}$$

Prediction learning using Functions (1) or (2) is limited with respect to changes in shape (see Chapter 5.3 of [18]). The problem can be expressed by a product of several probability densities over the rigid body transformation, encoding global as well as local contact configurations. Figure 2 shows the frames representing two different *experts* [18].

$$\begin{aligned} p_{\text{global}}(T^{B_t,B_{t+1}} | T^{A_t,A_{t+1}}, T^{A_t,B_t}, T^{B_t,O}) \\ p_{\text{local}}(T^{B^l_t,B^l_{t+1}} | T^{A^l_t,A^l_{t+1}}, T^{A^l_t,B^l_t}) \end{aligned} \tag{3}$$

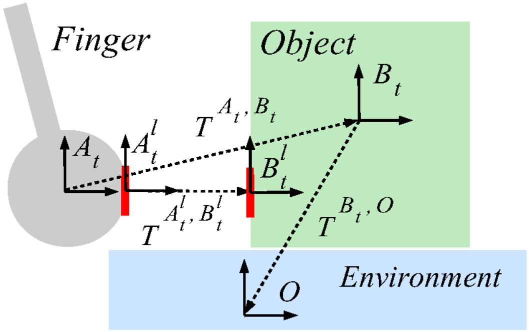


Fig. 2. A robotic finger A pushing an object B in an environment O, decomposed into local and global coordinate frames.

In addition to learning how an object moves in response to a push, it is desirable if we can also incorporate learned information about the inherent tendencies of parts of an object to move in various directions with respect to the environment or any other objects, but regardless of whether it is being pushed or not. This additional information may help when predicting the motions of previously unseen objects, because it provides some prior knowledge about what kinds of motions are possible and which are not.

The subsequent motion of the object in the inertial frame can now be described as:

$$\begin{aligned} p(T^{B_t,B_{t+1}}|K) &= \\ p_{\text{local}}(T^{B_t,B_{t+1}}|T^{A^l_t,A^l_{t+1}},T^{A^l_t,B^l_t}) &\times \\ p_{\text{global}}(T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}},T^{A_t,B_t},T^{B_t,O}) & \end{aligned} \tag{4}$$

where K stands for the known conditions $T^{A^l_t,A^l_{t+1}}$, $T^{A^l_t,B^l_t}$, $T^{A_t,A_{t+1}}$, T^{A_t,B_t} and $T^{B_t,O}$. The density product (4) is maximised using the differential evolution optimization algorithm [19].

Now the prediction system can be trained by training each of the density terms of Equation (4) on data derived simply by observing the outcomes of random robotic pushes, and extracting the resulting object motions with the tracker presented in the next Section.

V. Tracking

Visual observation of the trajectory of the object is the problem of finding the pose $T^{B_t,O}$ given an image I where the camera is assumed to be fixed to the global environment O. We are proposing a probabilistic framework and therefore searching for the conditional probability distribution $p(T^{B_t,O}|Z)$. Estimation is essentially based on Bayes' theorem:

$$p(T^{B_t,O}|Z) = \frac{p(Z|T^{B_t,O})p(T^{B_t,O})}{p(Z)} \tag{5}$$

where Z is a set of observations. $p(Z)$ and $p(T^{B_t,O})$ are assumed to be uniformly distributed, which means that the unconditioned probability of any observation or pose is the same, and thus Equation (5) simplifies to

$$p(T^{B_t,O}|Z) \propto p(Z|T^{B_t,O}) \tag{6}$$

For estimating this distribution we are using the methods proposed in [17, 20]. The basic idea is to project a texture based representation of the object into the current camera image and to maximise the match using a particle filter.

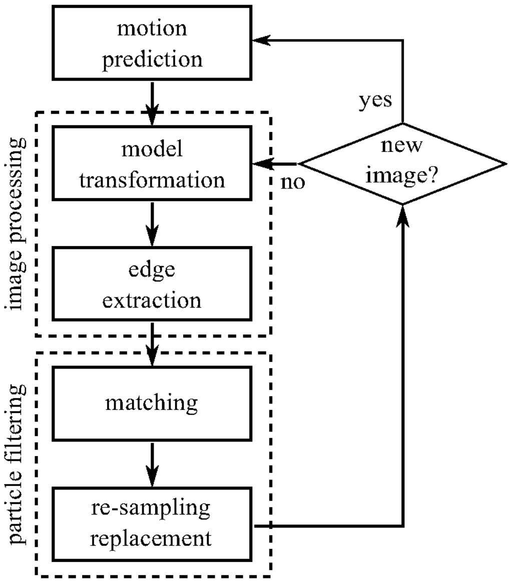


Fig. 3. Flow chart of the tracking algorithm using motion prediction and iterative particle filtering.

A. Particle filter

A particle filter, such as the SIR (Sequential Importance Resampling) as explained in [21], estimates the current state $\mathbf{x}_{t+1}$ based on the previous state $\mathbf{x}_t$ and the current observation z_{t+1}, where the system state is perturbed by system noise $\mathbf{n}_{t+1}$ and the observation by observation noise v_{t+1}. The particle filter maintains a probability distribution over the system state represented as a set of samples.

$$\begin{array}{rclcl} \mathbf{x}_{t+1} & = & f(\mathbf{x}_t, \mathbf{u}_t) & + & \mathbf{n}_{t+1} \\ z_{t+1} & = & h(\mathbf{x}_{t+1}) & + & v_{t+1} \end{array} \tag{7}$$

In our case the system state to be estimated is the transformation $T^{B_{t+1},O}$. $f(\mathbf{x}_t, \mathbf{u}_t)$ corresponds to function (2) where $\mathbf{u}_t \equiv K$. In the case of not using prediction at all Equation (7) simplifies to

$$\begin{array}{rclcl} \mathbf{x}_{t+1} & = & \mathbf{x}_t & + & \mathbf{n}_{t+1} \\ z_{t+1} & = & h(\mathbf{x}_{t+1}) & + & v_{t+1} \end{array} \tag{8}$$

i.e. $f(.)$ reduces to a static motion model.

The observation is given by Equation (12) and (13) in Section V-C respectively, i.e. the probability of the observation of a sample i is taken to be

$$p(z|T_i^{B_{t+1},O}) \propto \exp{(w_i)} \tag{9}$$

Note that the term *sample* is equivalent to *particle*, *pose* and *transformation* with respect to $T^{B_{t+1},O}$.

System noise $\mathbf{n}$ is defined as

$$\begin{array}{rcl} \mathbf{n} & = & N(0,\sigma) \\ \sigma & = & (1-c)\sigma_0 \\ c & = & \mathrm{mean}(w_i) \end{array} \quad (10)$$

where c is the confidence of the particle distribution in the current frame. Given the requirements for tracking accuracy and speed for a typical table top scenario we chose a basic standard deviation σ_0 of 0.03 m for the translational and 0.5 rad for the rotational degrees of freedom. The standard deviation is then scaled based on the confidence. This means that as the confidence of the particles increases its noise level decreases, leading to faster convergence. Observation noise v is modeled in a similar manner as in Equation (10) with a σ_0 of 4 pixels.

The particle filter always tries to find the local maximum in observation space. In the case of occlusion, as shown in Figure 6 and 8, this leads to drifting of the tracker away from the real pose of the object. To cope with this problem we introduce a threshold c_{th} for the confidence value c setting the system noise level such that

$$\begin{array}{rll} \mathbf{n} & = N(0,\sigma) & \text{if} \quad c > c_{th} \\ \mathbf{n} & = 0 & \text{if} \quad c \leq c_{th} \end{array} \quad (11)$$

This means that below a certain confidence $c \leq c_{th}$ the tracker relies completely on the output of the predictor. We currently set this threshold to an empirically determined value between 0.3 and 0.5.

B. Image processing

We project the geometric model of the object (described by vertices, faces and textures) into the image space using the transformation $T^{B_t,O}$ and standard techniques of computer graphics such as perspective transformation and texture mapping. In image space we compute the edge gradients of the model $\mathbf{g}_M$ and of the image captured by the camera $\mathbf{g}_I$.

C. Matching

Now it is possible to find a measure for the match, or weight w of a pose $T^{B_t,O}$. For each point (u,v) on the model M in image space we can compute the deviation of the edge gradients by superimposing the projected model over the image. The sum of the difference of the gradients is computed as:

$$\begin{array}{rcl} w_i & \propto & \frac{1}{e_i} \int_{(u,v)\in M} |\mathbf{g}_M(u,v) - \mathbf{g}_I(u,v)| \\ e_i & = & \int_{(u,v)\in M} |\mathbf{g}_M(u,v)| \end{array} \quad (12)$$

where $\mathbf{g}_M$ and $\mathbf{g}_I$ are the colour gradients of the projected model and the image respectively. A second approach for matching is similar to (12), with respect to pixel-wise comparison of the model and the image. But instead of computing the difference of gradients, the difference of the colour with respect to the hue in HSV (Hue, Saturation, Value) colour space is used.

$$w_i \propto \frac{1}{M} \int_{(u,v)\in M} |h_M(u,v) - h_I(u,v)| \quad (13)$$

where h_M and h_I are the hue values of the projected model and the image respectively. The advantage of using a colour based tracker is increased robustness against edge based clutter as in Figure 8. Of course it is less robust against changing lighting but the combination of both kinds of cues can significantly improve the overall performance. How to combine the two methods in an optimal way is an open issue and remains as future work.

Figure 3 gives an overview of our method. As proposed in [17] and [20], we use iterative particle filtering for better computational performance and accuracy of the tracker. To initialise the pose of the object we used SIFTs for visual recognition and RANSAC for 3D pose matching. For more details on this method please look up Section V in [22].

VI. Results

In all our experiments we are using the tracking system as described in Section V, and compare it against the tracker without prediction which is the same system but without the *motion prediction* step as in Figure 3. Other than *motion prediction* we are using the same configuration for each of the tracker, respectively non-iterative particle filtering with 100 particles for each frame. The number of particles was chosen small enough to ensure real-time operation in normal conditions, which however meant the tracker would run into problems as conditions deteriorate. One option in such a case would be to increase the number of particles, accepting loss of real-time performance (and e.g. buffering images), and indeed the tracker allows such dynamic resizing of the particle set. The approach taken in this paper however is to rely on an improved motion model based on the learned predictors rather than throw more particles at the problem, which allows us to also cover very severe cases where no number of particles can maintain a successful track. The following experiments are designed to illustrate the differences in performance between tracking with and without incorporating a learned prediction system. The poses are drawn as wireframe models with the following colour-code:

- *White*: Ground truth. Note that we did not use an external system such as a magnetic tracker for obtaining ground truth, but used the visual tracker itself in a high accuracy non real-time setting with many iterations and particles (4 and 200 respectively).
- *Green*: Tracking with prediction as proposed in this paper.
- *Red/Magenta*: Tracking without motion prediction.
- *Blue*: Pure motion prediction without visual feedback by the tracker, i.e. in Equation (7) $\mathbf{n}_{t+1} = 0$.

For visual observation a camera is capturing images with a resolution of 800×600 at a frame rate of 30 Hz and highly accurate time-stamps. Tracking is executed in real-time whereas in critical situations, where the prediction system has to take over, the data is buffered and evaluated at a frame rate of 1-5 Hz.

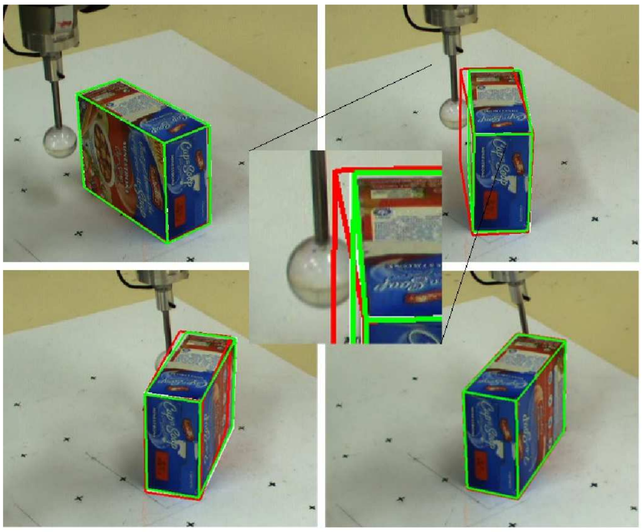


Fig. 4. Overlaid edge-based tracking results with and without prediction, from top-left to bottom-right. (ground truth: white, tracking with prediction: green, tracking without prediction: red)

A. Experiment - Accuracy

In this experiment we show how accuracy increases using the proposed methods. We compare the poses of tracking with and without prediction against ground truth. To evaluate the error in both cases we used the non-normalized error measure described in Section IV-B of [23].

$$E_t = \sum_{n=1...N} |p_n^{2,t} - p_n^{1,t}| \tag{14}$$

where $p_n^{1,t}$ are randomly chosen points on the object surface at the ground truth pose. Thus E_t measures the mean displacement of the object surface.

For a fair comparison we only used pushing examples, where also the tracker without prediction was able to maintain tracking throughout the whole sequence, as shown in Figure 4. In the upper-left image both trackers are initialised at the same pose. The upper-right and lower-left show the mismatch of the tracker without prediction (*red*) while the object is moving. At the end of the sequence, lower-right image, both trackers converge to the ground-truth pose as to be expected.

Figure 5 shows the result of the evaluation of 20 pushes. For both, average error and standard deviation, the tracker which takes advantage of information from the predictor performs significantly better.

The following experiments show robustness to various events in a qualitative manner. For these cases a quantitative evaluation against the tracker without prediction is meaningless, as the latter loses the object at some point altogeher.

B. Experiment - Occlusion

Typically visual tracking algorithms fail when the object is partially or completely occluded. The prediction model of Section IV allows us to overcome such situations. Figure 6 shows a pushing sequence which suffers from heavy occlusion. At the beginning both of the trackers perform very well, since at least parts of the object are visible (top row). At the point where the occluder is completely hiding the object, the tracker without predictor (magenta wireframe) is drifting away to a visually more likely position (e.g. it is attracted to the robot hand which introduces clutter with respect to edges as well as colour) and fails to keep track of the object pose.

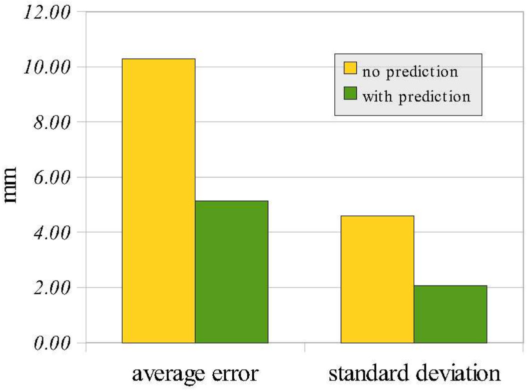


Fig. 5. Tracking accuracy with and without prediction: average displacement error of surface points.

Pure prediction (blue wireframe) does not use any visual feedback and produces the whole trajectory from the initial pose. The finger is pushing very near to the centre of the object which is a very unstable position. Given this push, the object in some cases might slide to the left or to the right, or topple over depending on slightly different initial positions. However, since the predictor used for tracking gets updated by the visual observation it is possible to handle such difficult situations.

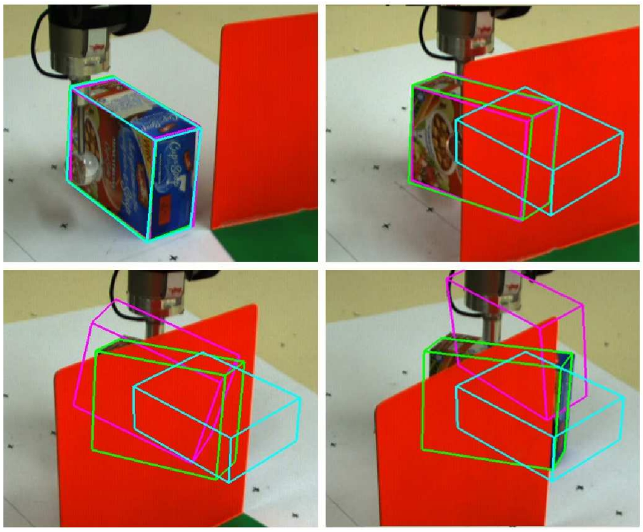


Fig. 6. Failure of tracking without prediction in case of occlusion. (edge-based tracking with prediction: green, without prediction: magenta, pure prediction: blue)

C. Experiment - Motion blur

Another example of a difficult situation is fast movement of the object relative to the camera. Figure 7 illustrates such a case, where a box is pushed forward causing it to tilt until it reaches an unstable pose and finally toppling over. This is a very critical situation for visual observation. The falling object moves quite fast, causing the image to blur.

Again the tracker with (green) and without prediction (red), and the pure predictor (blue) are initialised at the same starting pose. During the first phase of the sequence the predictor proposes an erroneous rotation of the object, while the vision system extracts the correct pose relatively accurately (top row of Figure 7). However, by the time of the unstable pose shown in the lower-left image the tracker with prediction is already better than the tracker without prediction. The object is moving fast during the next frames causing the effects mentioned above. The tracker without prediction can not follow the fast movement, loses track and gets trapped in a local maximum. The predictor on the other hand proposes the right pose and the corresponding tracker refines the result.

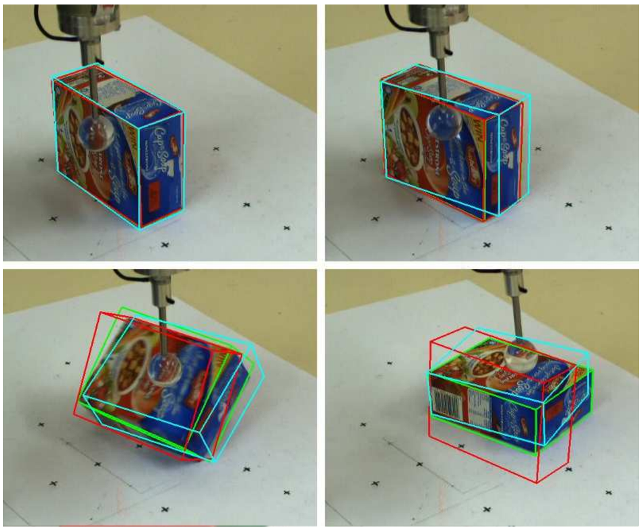


Fig. 7. Failure of tracking without prediction in case of a toppling object. (edge-based tracking with prediction: green, without prediction: red, pure prediction: blue)

D. Experiment - Occlusion with motion blur

The hardest case for a visual observation system is the combination of fast movement and occlusion. We tested this case by applying a pushing manipulation where the object is hidden behind an occluder where it topples over, as shown in Figure 8.

In the top-left image enough parts of the object are visible and both of the trackers produce good results. The top-right image shows the object behind the occluder already in the phase of falling down as the blur suggests. The lower-left is the subsequent frame and illustrates the large change of the pose, which causes the tracker without prediction to fail, whereas the tracker with prediction overcomes this difficult situation. The pose of the pure predictor is also very close to the real one, but suffers from integrating error over the trajectory.

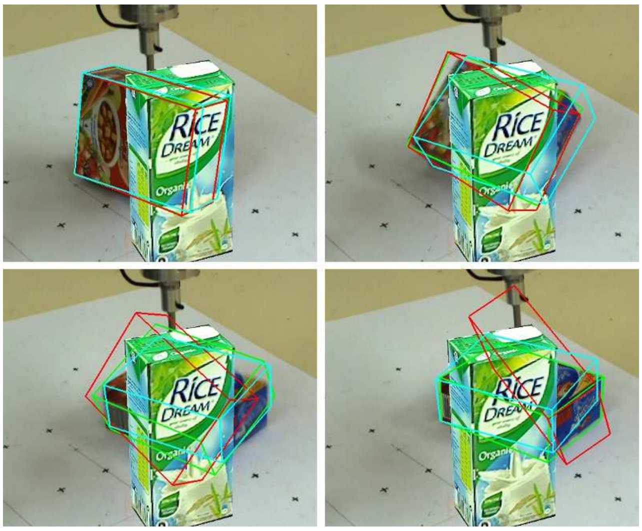


Fig. 8. The toughest case: toppling combined with occlusion. (colour-based tracking with prediction: green, without prediction: red, pure prediction: blue)

Note that for this experiment we placed a virtual occluding object in the scene. This allowed us to vary the size and texture of the occluder and most importantly to position it right in front of the toppling object.

VII. Conclusions and Future Work

A. Conclusions

In this work we demonstrated how objects can be tracked in 3D under visually challenging situations such as occlusion, motion blur and fast movement. We summarized the main ideas of probabilistic prediction and explained edge and colour based 3D tracking using a Monte Carlo particle filter. We show how the use of probabilistic prediction as motion model for the tracker leads to clearly improved accuracy as well as robustness. There are cases where erroneous prediction can degrade tracking performance (e.g. in unstable contact configurations), but these are outweighed by the majority of cases where prediciton is correct and especially for occlusion.

B. Future Work

Although the robustness of the tracker is already improved, there are several points which remain open.

First of all the confidence of a visual observation is not very distinctive and it is very hard to find a good measure indicating whether the pose it suggests is wrong or not. As described in Section V we manually set a threshold. Obviously this is not the optimal solution since this threshold depends on the visual model of the object and the clutter in the environment, especially introduced by the occluder.

I.e. for an edge based tracker it is very hard to tell if it is correct or not in case of edge rich clutter as shown in Figure 8, whereas the colour based tracker is strongly influenced by objects which are coloured similar to the object, with respect to Equation (13).

Although the predictor is trained using visual tracking, this still happens in an offline stage for now. For an automated system it is desirable to learn the predictor online. This leads to a chicken-and-egg problem. Since at the beginning the predictor is not trained, the tracker produces bad results for challenging situations, which can thus not be learned by the predictor. As mentioned above, the tracker is not able to reliably measure the correctness of the pose suggested and the predictor will thus learn those wrong poses.

Furthermore, at the moment the predictor provides only the most likely pose in Equation (4) to the tracker. This means that we do not make full use yet of the probabilistic framework inside the predictor.

VIII. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement No. 215181, CogX.

References

[1] T. Drummond and R. Cipolla, "Real-time tracking of multiple articulated structures in multiple views," in *European conference on Computer Vision*, 2000.

[2] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online grasp planning," in *IEEE International Conference on Robotics and Automation, ICRA*, 2001.

[3] C. Harris and A. Blake ed., "Tracking with rigid bodies," *Active Vision*, pp. 59–73, 1992.

[4] D. Lowe, "Robust model-based motion tracking through the integration of search and estimation," in *International Journal of Computer Vision*, pp. 113–122, 1992.

[5] P. Wunsch and G. Hirzinger, "Registration of cad-models to images by iterative inverse perspective matching," in *Proceedings of the 13th International Conference on Pattern Recognition*, pp. 77–83, 1996.

[6] G. Klein and D. Murray, "Full-3d edge tracking with a particle filter," in *Proc 17th British Machine Vision Conference*, 2006.

[7] J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "Gpu-accelerated real-time 3d tracking for humanoid locomotion," in *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[8] E. Murphy-Chutorian and M. M. Trivedi, "Particle filtering with rendered models: A two pass approach to multi-object 3d tracking with the gpu," in *Computer Vision and Pattern Recognition Workshop*, 2008.

[9] M. Mason, "Manipulator grasping and pushing operations," in *PhD thesis MIT*, 1982.

[10] M. Peshkin and A. Sanderson, "The motion of a pushed, sliding workpiece," in *IEEE Journal on Robotics and Automation*, vol. 4, pp. 569–598, 1988.

[11] K. Lynch, "The mechanics of fine manipulation by pushing," in *IEEE International Conference on Robotics and Automation*, pp. 2269–2276, 1992.

[12] Y. Aiyama, M. Inaba, and H. Inoue, "Pivoting: A new method of graspless manipulation of object by robot fingers," in *International Conference on Intelligent Robots and Systems, Proceedings of the 1993 IEEE/RSJ*, pp. 136 –143 vol.1, 1993.

[13] K. M. Lynch, "Toppling manipulation," in *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 152–159, 1999.

[14] D. Cappelleri, J. Fink, B. Mukundakrishnan, V. Kumar, and J. Trinkle, "Designing open-loop plans for planar micro-manipulation," in *IEEE International Conference on Robotics and Automation*, pp. 637–642, 2006.

[15] D. Duff, J. Wyatt, and R. Stolkin, "Motion estimation using physical simulation," in *IEEE International Conference on Robotics and Automation*, 2010.

[16] M. Kopicki, J. Wyatt, and R. Stolkin, "Prediction learning in robotic pushing manipulation," in *International Conference on Advanced Robotics*, pp. 1–6, 2009.

[17] T. Mörwald, M. Zillich, and M. Vincze, "Edge tracking of textured objects with a recursive particle filter," in *Proceedings of the Graphicon 2009*, (Moscow, Russia), 2009.

[18] M. Kopicki, *Prediction learning in robotic manipulation*. PhD thesis, University of Birmingham, 2010.

[19] R. Storn and K. Price, "Differential evolution. a simple and efficient heuristic for global optimization over continuous spaces," in *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[20] A. Richtsfeld, T. Mörwald, M. Zillich, and M. Vincze, "Taking in shape: Detection and tracking of basic 3d shapes in a robotics context," in *Computer Vision Winter Workshop*, pp. 91–98, 2010.

[21] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[22] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "Blort - the blocks world robotic vision toolbox," in *Proceedings of the ICRA workshop*, 2010.

[23] M. Kopicki, R. Stolkin, S. Zurek, T. Mörwald, and J. Wyatt, "Predicting workpiece motions under pushing manipulations using the principle of minimum energy," in *Proceedings of the RSS workshop*, 2009.

Assessing Grasp Stability Based on Learning and Haptic Data

Yasemin Bekiroglu, Janne Laaksonen, Jimmy Alison Jørgensen, Ville Kyrki, *Member, IEEE*, and Danica Kragic, *Member, IEEE*

***Abstract*—An important ability of a robot that interacts with the environment and manipulates objects is to deal with the uncertainty in sensory data. Sensory information is necessary to, for example, perform online assessment of grasp stability. We present methods to assess grasp stability based on haptic data and machine-learning methods, including AdaBoost, support vector machines (SVMs), and hidden Markov models (HMMs). In particular, we study the effect of different sensory streams to grasp stability. This includes object information such as shape; grasp information such as approach vector; tactile measurements from fingertips; and joint configuration of the hand. Sensory knowledge affects the success of the grasping process both in the planning stage (before a grasp is executed) and during the execution of the grasp (closed-loop online control). In this paper, we study both of these aspects. We propose a probabilistic learning framework to assess grasp stability and demonstrate that knowledge about grasp stability can be inferred using information from tactile sensors. Experiments on both simulated and real data are shown. The results indicate that the idea to exploit the learning approach is applicable in realistic scenarios, which opens a number of interesting venues for the future research.**

***Index Terms*—Force and tactile sensing, grasping, learning and adaptive systems.**

I. Introduction

GRASPING is an essential skill for a general-purpose service robot, working in an industrial or home-like environment. If object parameters such as pose, shape, weight, and/or material properties are known, grasp planning that uses analytical approaches can be employed [1]. In unstructured environments, these parameters are uncertain, which present a great challenge for the current state-of-the-art approaches.

Extraction and appropriate modeling of sensor data can alleviate the problem of uncertainty. Many approaches to robotic object grasping exist, and most of these have been designed to deal with known objects. To estimate the shape and pose of an object, visual sensing has been widely used [2]–[7]. However, the accuracy of vision is limited, for example, due to imperfect calibration and occlusions. Small errors in object pose are, thus, common even for known objects, and these errors may cause failures in grasping. These failures are commonly difficult to prevent at the grasp execution stage if the hand is not equipped with sensors. Tactile and finger force sensors can be used to reduce some problems [8], [9] but are still uncommon in practice. We have observed that due to uncertainty in the observations, a grasp may fail due to slippage or collision, even when all fingers have adequate contact forces, and the hand pose with respect to the object is not very different from the planned one.

The main contribution of our study is a new approach that incorporates knowledge of uncertainty in the observations when predicting the stability of a grasp. We show how grasp stability can be assessed based on data extracted both *prior to* and *during* execution. The data contain object information such as shape, grasp information such as approach vector, and online sensory and proprioceptive data including tactile measurements from fingertips and joint configuration of the hand. In a real-world robot platform, all measurements that are acquired from the environment are noisy and associated with a degree of uncertainty. Our goal is to create a system which is capable of performing prediction of grasp stability from real-world sensory streams. In order for the system to be robust, the uncertainty in the observations needs to be taken into account. Probabilistic methods provide a framework to deal with uncertainty in a principled manner and will, to this end, provide the foundation on which our system is built. Our aim is to model the embodiment specific and inherently complex relationship between grasp stability and the available sensory and proprioceptive information. Methods that are based on AdaBoost, support vector machines (SVMs), and hidden Markov models (HMMs) are proposed and compared.

Our approach is a learning-based framework that relies on having a training dataset that is assumed to sample the domain of possible scenarios well. This poses a challenge: Acquiring such data is associated with a significant cost with respect to time and computation. In order to alleviate this problem, we use a simulator from which we can generate a large set of synthetic training data in a controlled environment with relative ease. The approach to use synthetic training data is justified by performing inference on real-world examples. Moreover, the generalizability of the grasp stability estimation is experimentally evaluated. The results demonstrated that the stability estimation

Manuscript received January 12, 2011; revised January 12, 2011; accepted March 15, 2011. Date of publication May 2, 2011; date of current version June 9, 2011. This paper was recommended for publication by Guest Editor G. Metta and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported in part by the European Union under Projects CogX, FP7-IP-027657, and GRASP, FP7-IP-215821 and in part by the Swedish Foundation for Strategic Research.

Y. Bekiroglu and D. Kragic are with the Centre for Autonomous Systems and Computer Vision and Active Perception Laboratory, School of Computer Science and Communication, Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: yaseminb@kth.se; danik@nada.kth.se).

J. Laaksonen and V. Kyrki are with the Department of Information Technology, Lappeenranta University of Technology, FIN-53850 Lappeenranta, Finland (e-mail: jalaakso@lut.fi; kyrki@lut.fi).

J. A. Jørgensen is with Robotics Group, Maersk Mc-Kinney Moller Institute, University of Southern Denmark, DK-5230 Odense, Denmark (e-mail: jimali@mmmi.sdu.dk).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2132870

generalizes well to new objects even with a moderate number of objects used in training. In summary, this paper demonstrates that knowledge about grasp stability can be inferred using information from tactile sensors, while grasping an object before the object is further manipulated. This is very useful since, if an unstable grasp is predicted, objects can be regrasped before attempting to further manipulate them.

In the following section, the contributions of our work are discussed in detail in relation to the state-of-the-art work in the area. This is followed by a presentation of the theoretical framework in Section III and the employed learning methodology. In Section IV, the simulator, the database, and the real-data collection are described. We present the results of experimental evaluation in Section V and conclude our work in Section VI.

II. Contributions and Related Work

In robotic object grasping, there has been a lot of effort during the past few decades [1]. Grasp stability analysis is a tool that is often used in grasp planning, where the grasp is planned using grasp quality measures derived from stability analysis. Most of the work on grasp stability assessment relies on analytical methods and focuses on rigid objects, albeit some work has considered the analysis of grasps on deformable objects [10]. Compared with our approach, the analytical methods require exact knowledge of the contacts between the hand and the object to estimate the stability of a grasp.

Most of the grasp-planning approaches that are tested in simulation have the common property to use a strategy that relies on the object shape. Modeling object shape with a number of primitives such as boxes, cylinders, cones, spheres [4], [11], or superquadrics [12] reduces the space of possible grasps. The decision about the suitable grasp is made based on grasp quality measures given contact positions. However, none of these approaches provide a principled way to deal with uncertainties that arise in dynamic scenarios or in the errors inherent to simplification with primitives, which can potentially be solved using tactile feedback. This is also the main objective and contribution of the study presented here.

One of the issues that are often faced in household scenarios is deformable objects. Planning grasps for these types of objects is not at all well studied as rigid objects. Examples can be found in the literature, such as [13], where the deformation properties of objects are learned, and then a suitable grasping force is planned for the associated objects.

To cope with the fact that the exact knowledge of the object and the hand is not available, we employ tactile sensors that measure a range of pressure levels. Tactile sensing has been used for various purposes in prior studies, and we focus on the use of tactile sensors in the remaining survey of the related work. There are recent examples that perform grasp generation from visual input and use tactile sensing for closed-loop control once in contact with the object. For example, the use of tactile sensors has been proposed to maximize the contact surface for removing a book from a bookshelf [14]. Application of force, visual, and tactile feedback to open a sliding door has been proposed in [15]. In our study, the main difference is that the tactile sensors are used to assess the stability of a grasp. Thus, rather than using the tactile data for control, we use them in order to reason about grasp stability.

Learning aspects have been considered in the context of grasping mostly for the purpose of understanding human grasping strategies. In [16], it was demonstrated how a robot system can learn grasping by human demonstration using a grasp experience database. The human grasp was recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup table. Another approach is to use vision. However, measuring the contact between object and hand accurately is a nontrivial task. The system in [2] learns grasping points by using hand-labeled training data in the form of image regions which indicate good grasping regions. A probabilistic decision system is employed on previously unseen objects to determine a good grasping point or region. In [3], vision is used to create grasp affordance hypotheses for objects and refine the grasp affordance hypotheses through grasping. The result is a set of grasps that will produce good grasps on a specific object.

Current learning approaches that use tactile sensors are focused on either determining the properties of objects [17]–[19] or object recognition [19]–[22]. Different properties of objects give valuable information that can be further used in grasp stability analysis. In [17], the pose of the object is determined using a particle-filter technique based on the tactile information gained from the contacts between a gripper and the object. Similar work was presented by Hsiao *et al.* [23], where object localization was performed with knowledge of tactile contacts on specific objects. In [18], the surface type (edge, flat, cylindrical, and sphere) of the tactile contact is determined using a neural network. In [19], tactile information that is extracted from the sensors on a two-fingered gripper is used to determine the deformation properties of an object. However, learning or analyzing such object properties through tactile sensors do not answer the question of grasp stability directly compared with the work presented here.

Work on using tactile sensors for recognition of manipulated objects has been reported rather recently. The main approach is to use multiple grasp or manipulation attempts and then learn the object through the haptic input from the manipulations or grasps. Current approaches use either one-shot data from the end of the grasps [21], [22] or temporal data collected throughout the grasp or manipulation execution [19], [20]. In [21], a bag-of-words approach is presented that aims to identify objects using touch sensors available on a two-fingered gripper. The approach processes tactile images collected by grasping objects at different heights. In [22], a similar approach is taken for a humanoid hand. A more traditional approach to learning is employed with features extracted from tactile images in conjunction with hand joint configurations as input data for the object classifier. In [20], entropy is used to study the performance of various features in order to determine the most useful features in recognizing objects. In this case, a plate that was covered with a tactile sensor was used as the manipulator. However, the object recognition using the recognized good features did not perform here as well as it did in the other presented works. Thus, no attempts have been made to use tactile sensors that are placed on a robotic hand to predict the stability of a grasp. We have presented the idea of grasp stability prediction using tactile sensors in [24] with some initial results, and we extend our work in this paper.

III. Problem Formulation and Modeling

To determine grasp stability is difficult, when factors that affect the stability are uncertain or unknown. We show that with a probabilistic approach, it is possible to assess grasp stability using tactile measurements. Mapping from tactile sensor measurements to grasp stability is complex and not injective because of variability in object parameters, grasp, and hand types, as well as the uncertainty inherent in the process. Thus, we consider grasp stability as a probability distribution

$$P(S|H(t), j(t), O, G) \tag{1}$$

where grasp stability, which is denoted by S, depends on different measured and/or known factors. The factors that are taken into account in our model are 1) H: force/pressure measurements from tactile sensors; 2) j: joint configuration of the hand; 3) O: object information, e.g., object identity or shape class; and 4) G: information relevant to the grasp, e.g., approach vector and/or hand preshape. Grasp stability S is a discrete variable with two possible states: a grasp is either stable or unstable, while the other variables can be discrete or continuous. Our goal is to assess the effect of factors in (1) to grasp stability by considering different subsets of the variables.

We study the problem using both instantaneous measurements of variables and time-series measurements. With instantaneous measurements, the stability is assessed only from the instant the robot hand is static and closed around the object. This approach is referred to as one-shot classification. In contrast, the time-series approach takes into account measurements that are generated during the whole grasping sequence. The variables H and j are, thus, represented from time t_0 to t_n, where t_0 and t_n represent the start and the end of the grasping sequence, respectively. In the case of one-shot classification, we use the measurements once the hand has reached a static configuration, which is an approach similar to [21]. Thus, we compare the distribution defined by (1) with one that discards the time series:

$$P(S|H(t_n), j(t_n), O, G). \tag{2}$$

We show that both approaches that are described by (1) and (2) are valid and that grasp stability can be assessed based on them. To study the contribution of object O and grasp knowledge G, we have set up a hierarchy as depicted in Fig. 1. The hierarchy is divided into levels, each with increasing amount of sensory information being available. At the top level of the hierarchy, only the information that is related to the hand itself, H, and j is used. Thus, we estimate

$$P(S|H, j) = \int\int P(S|H, j, O, G)\, p(G|O)\, p(O)\, dO\, dG\,. \tag{3}$$

Considering only sensor information, the overall distribution will be somewhat uninformative—there is significant uncertainty as the same sensor readings can be associated with both stable and unstable grasps for different objects, grasp approach vectors, and hand preshapes. Subsequently, when more pieces of information are considered, the estimation of the distribution should be more specific, resulting in better discrimination. At the second level, we consider that object shape or object instance is known:

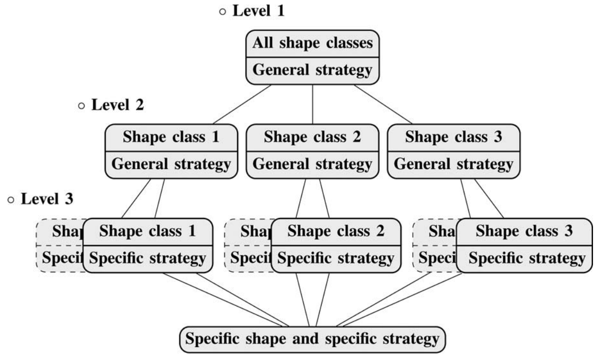


Fig. 1. Hierarchical recognition of grasp stability taking into account different types of sensory knowledge.

$$P(S|H, j, O) = \int P(S|H, j, O, G)\, p(G)\, dG\,. \tag{4}$$

Finally, at the third level, we consider knowledge about the applied grasp, and estimate the stability through $P(S|H, j, O, G)$. Since knowledge of all the variables that are present in (1) is assumed, the uncertainty in the stability estimation is expected to decrease.

In the rest of this section, we describe methods to estimate the density functions using a classification approach. SVMs and AdaBoost are used to model the instantaneous model, according to (2), while HMMs are used for the general time-series case, according to (1). Although the probabilistic framework is presented as a method to estimate grasp stability using haptic data, it is also possible to use the proposed framework with other types of sensory information.

A. Feature Representation

First, we describe the input features for the classifiers. In this work, a three-fingered Schunk Dextrous Hand (SDH) with seven degrees of freedom and equipped with six 2-D Weiss Robotics pressure-sensitive tactile pads [25] is used as a demonstration hardware platform. Tactile measurements are recorded from the first contact with the object until a steady state is reached. The whole measurement sequence is denoted by $x_1^i, \ldots, x_{T_i}^i$, where i is the index of the measurement. For one-shot classification, tactile measurements at the steady state are used and denoted $x_{T_i}^i$. Training data are generated both in simulation and on real hardware and will be presented in Section IV. The notation used in this paper is as follows.

1) $D = [o_i], i = 1, \ldots, N$ denotes a dataset with N observation sequences.
2) $o_i = [x_t^i], t = 1, \ldots, T_i$ is an observation sequence.
3) $x_t^i = [M_f^{i,t} j_v^{i,t}], f = 1, \ldots, F, v = 1, \ldots, V$ is the observation at time instant t given the ith sequence; F is the

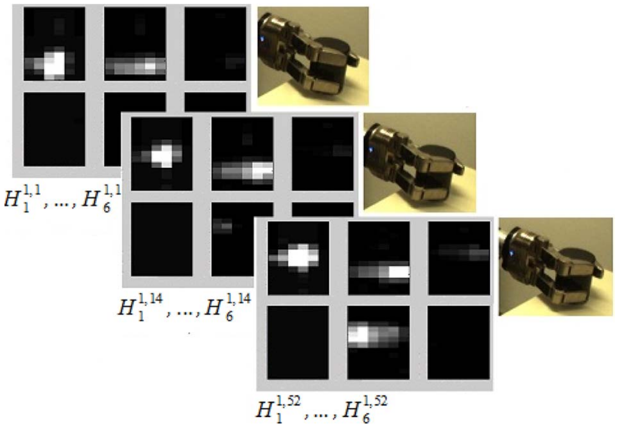

$H_1^{1,1}, \ldots, H_6^{1,1}$
$H_1^{1,14}, \ldots, H_6^{1,14}$
$H_1^{1,52}, \ldots, H_6^{1,52}$

Fig. 2. Example grasping sequence of a cylinder and the corresponding tactile measurements.

number of tactile sensors; and V is the number of joints of the robot hand.

4) $M_f^{i,t}$ includes the moment features that are extracted from the tactile readings $H_f^{i,t}$ on the sensor f at time instant t, given the ith sequence. Details about the extraction of these features are given later in this section.

5) $j_v^{i,t}$ is a joint angle at time instant t given the ith sequence.

The acquired data, thus, consist of tactile readings $H_f^{i,t}$ and joint angles of the hand $j_v^{i,t}$. For the SDH, we store $3 \times (14 \times 6)$ readings on proximal and $3 \times (13 \times 6)$ on distal sensors, and seven parameters represent the pose of the hand given the joint angles. Example images from the tactile sensors are shown in Fig. 2. The tactile images in the figure represent a stable grasp of a cylinder.

Tactile data are relatively high dimensional and redundant. Thus, we borrow ideas from image processing and consider the 2-D tactile patches as images. Each tactile image is represented using image moments. The general parameterization of image moments is given by

$$m_{p,q} = \sum_z \sum_y z^p y^q f(z, y) \qquad (5)$$

where p and q represent the order of the moment, z and y represent the horizontal and vertical positions on the tactile patch, respectively, and $f(z, y)$ represents the measured contact. We compute moments up to order 2, $(p + q) \in \{0, 1, 2\}$, for each sensor array separately. These then correspond to the total pressure and the distribution of the pressure in the horizontal and vertical directions. Thus, there are in total six features for each sensor resulting in an observation $x_t^i \in \mathbb{R}^{6F+V}$. Normalizing the feature vector is a common step in machine-learning methods. In our case, moment features and finger joint angles are normalized to zero mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

B. One-Shot Recognition

In this section, we examine the learning of grasp stability based on tactile measurements acquired at the end of a grasping sequence, i.e., once the final grasp has been applied to the object. We claim that if successful separation between stable and unstable grasps can be learned from examples, one-shot classification can determine the stability of the grasp from any haptic observation x_t^i measured during a grasp. This information can then be used in grasp control to determine when the robot hand has reached a stable configuration.

In this paper, two types of nonlinear classifiers, AdaBoost and SVM, are used in the experiments to demonstrate the ability to learn the stability of the grasps. AdaBoost and SVM were the best performing classifiers in [26]. AdaBoost is a boosting classifier, which has been developed by Freund and Schapire [27], that works with multiple so-called weak learners to form a committee that performs as the classifier. Here, we use AdaBoost implementation from [28].

SVM classification [29], [30] is also suitable for the problem. SVM is a maximum margin classifier, i.e., the classifier fits the decision boundary so that maximum margin between the classes is achieved. This guarantees that the generalization ability between the classes is not lost during the training of the SVM classifier. We use the libSVM implementation presented in [31]. Another critical feature of the SVM for our use is the ability to use nonlinear classifiers instead of the original linear hyperplane classifier. Nonlinearity is achieved using different kernels; in this study, the radial basis function

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \quad \text{for} \quad \gamma > 0 \qquad (6)$$

is used as the kernel for SVM. Moreover, as an extension to the basic two-class SVM, probabilistic outputs for SVM are used to analyze the results given by the SVM. This idea was first presented in [32]. The SVM output $y(\mathbf{x})$ is converted to a probability according to

$$p(t = 1|\mathbf{x}) = \sigma(\Gamma y(\mathbf{x}) + \Lambda), \qquad y(\mathbf{x}) = K(\mathbf{w}, \mathbf{x}) + b \qquad (7)$$

where parameters Γ and Λ are estimated using training data, and $\sigma(\cdot)$ is the logistic sigmoid function. This probability is, thus, related to the earlier general discussion by

$$P(S = \text{stable}|H(t), j(t), O, G) = p(t = 1|\mathbf{x}) \; . \qquad (8)$$

C. Temporal Recognition Using Hidden Markov Models

Time-series grasp stability assessment is performed based on HMMs [33]. Here, we use HMM implementation from [34]. We construct two HMMs: one representing stable and one unstable grasps. Classification of a new grasp sequence is performed by evaluating the likelihood of both models and choosing the one with higher likelihood. For the HMM, we use the notation $\lambda = (\pi, A, B)$, where π denotes the initial probability distribution, A is the transition probability matrix

$$A = a_{ij} = P(S_{t+1} = j|S_t = i), \qquad i, j = 1 \ldots N \qquad (9)$$

and B defines output (observation) probability distributions $b_j(x) = f_{X_t|S_t}(x|j)$, where $X_t = x$ represents a feature vector for any given state $S_t = j$. In this paper, we evaluate both ergodic (fully connected) and left-to-right HMMs.

The estimation of the HMM model parameters is based on the Baum–Welch procedure. The output probability distributions are modeled using Gaussian mixture models (GMMs)

$$f_X(x) = \sum_{k=1}^{K} w_k \frac{1}{2\pi^{L/2}\sqrt{|C_k|}} e^{-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)} \qquad (10)$$

where $\sum_{k=1}^{K} w_k = 1$, μ_k is the mean vector, and C_k is the covariance matrix for the kth mixture component. The unknown parameters $\theta = (w_k, \mu_k, C_k : k = 1...K)$ are estimated from the training sequences $o = (x_1, ...x_T)$. Initial estimates of the observation densities in (10) affect the point of convergence of the reestimation formulas. Depending on the structure of the HMM (ergodic versus left-to-right), we use a different initialization method for the parameters of the observation densities. The two initialization procedures are given as follows.

1) For an ergodic HMM, observations are clustered using k-means. Here, k is equal to the number of states in the HMM, and each cluster is modeled with a GMM using standard expectation maximization. Initial parameters for the GMMs are found using k-means algorithm.
2) For a left-to-right HMM, each observation sequence is divided temporally into equal length subsequences. Then, each GMM is estimated from the collection of corresponding subsequences. Thus, the GMMs represent the temporal evolution of the observations. Initial parameters are found as in the case of an ergodic HMM.

IV. Data Collection

For a learning system to achieve good generalization capabilities, relatively large training data are typically required. To generate large datasets on real hardware is time consuming, and in robotic grasping, it is difficult to generate repeatable experiments due to the dynamics of the process. However, if suitable models are available, simulation can be used for generation of data for both training the learning system and performance evaluation. In our study, we generate both simulated and real training data as explained next.

A. Simulator

The grasp simulator RobWorkSim,[1] which is described in [35], is used to generate training data including tactile measurements. The simulator is used in combination with the open dynamics engine (ODE) physics engine and provides support to simulate articulated hands, PD joint controllers, grasp quality measures, camera sensors, range sensors, and tactile sensors. The primary motivation to use RobWorkSim over the more widely used GraspIt! [36] is the integrated support for tactile array sensors.

1) Tactile sensor model: The tactile array sensor simulation in RobWorkSim is an experimental model that transforms the point contacts of the ODE to sensor measurements by describing the deformation of the sensor surface given a point force $\mathbf{f}$ applied perpendicular to it. The model was originally described in [37]. The model assumes that the deformation or response is linear with the magnitude of the point force, which is a fair assumption for small forces. Given the deformation function $h(x, y)$ where x and y are specified relative to the center (a, b) of the contact, the total deformation of the surface of an array of rectangular texels with size (A, B) can be found by integrating over the surface of each texel by

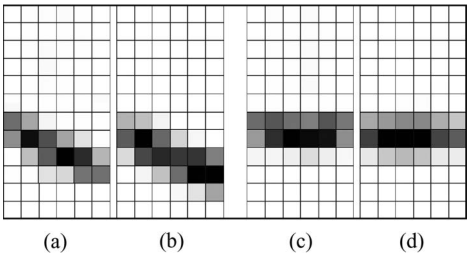


Fig. 3. Measured (a) and (c) versus simulated (b) and (d) sensor values. The tactile images were generated by pressing a sharp edge onto the sensor surface.

$$g_{m,n}(a, b) = \int_{(A-\frac{1}{2})m-a}^{(A+\frac{1}{2})m-a} \int_{(B-\frac{1}{2})n-b}^{(B+\frac{1}{2})n-b} h(x, y)dxdy \qquad (11)$$

where (a, b) is the center point of the contact, and (m, n) is the texel index. This surface integration is approximated using the rectangle method. Point force experiments on the real sensors suggested that the deformation decreased with the inverse of the square of the distance from the point force. We use an isotropic function to approximate the deformation of the sensor surface

$$h(x, y) = (\mathbf{f} \cdot \mathbf{n}_{\mathbf{texel}})\max\left(-\beta + \frac{\alpha}{1 + x^2 + y^2}, 0\right) \qquad (12)$$

where (x, y) is specified relative to (a, b), and $\mathbf{n}_{\mathbf{texel}}$ is the normal of the texel on which the point force $\mathbf{f}$ is applied. The parameters (α, β) were found by fitting the model to experimental data extracted from real sensors. Fig. 3 shows a visual comparison between the real and the simulated sensor output, where a sharp edge was pressed against both sensors.

Assessing grasp quality requires taking properties of the hand (orientation, joint configuration, friction, elasticity, and grasping force) and object (shape, mass, friction, contact locations, area, and contact force) into account. In the simulated environment, these parameters are known. We use a widely known grasp quality measure based on the radius ϵ of the largest enclosing ball in the grasp wrench space (GWS). We construct the GWS as proposed in [38] by calculating the convex hull over the set of contact wrenches $\mathbf{w}_{i,j} = [\mathbf{f}_{i,j}^T \lambda(\mathbf{d}_i \times \mathbf{f}_{i,j})^T]^T$, where $\mathbf{f}_{i,j}$ belongs to a representative set of forces on the extrema of the friction cone of contact i. $\mathbf{d}_i$ is the vector from the torque origin to contact i, and λ weighs the torque quality relative to the force quality.

It is not obvious how to determine λ due to the differences between forces and torques. We, therefore, calculate force space and torque space independently and use the radius of the largest enclosing ball in each of these to give a 2-D quality value $(\epsilon_f, \epsilon_\tau)$ for each grasp. A third quality measure ϵ_{cmc} that is based on

[1] More information about RobWorkSim can be found in `http : //www.robwork.org/`.

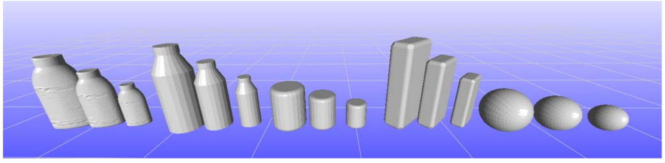


Fig. 4. Objects in simulation were generated in three sizes (75%, 100%, and 125%): hamburger sauce, bottle, cylinder, box, sphere.

the distance between the centroid of the contact polygon C and the center of mass CM of the object [39] is used: $\epsilon_{cmc} = \|CM - C\|$. This measure captures the same properties as the torque measure; however, it is more robust with regard to the point contact output of the simulator. Stable grasps are defined as those for which all three quality values are within a certain threshold. The thresholds have been determined experimentally.

B. Generating Training Data in Simulation

The database includes examples of stable and unstable grasps on different objects. We examine stability starting from the most general case in the hierarchy specified in Fig. 1 and continue by including information about subsequent properties until reaching the most specific case. At the top level of the hierarchy, data are generated on objects with different shapes using approach vectors that are generated uniformly from a sphere, referred to as a spherical strategy. At the second level, the shape information is given; hence, grasps are generated separately per object shape with the spherical strategy. At the third level, the approach vector is formed based on the object shape: Side or top grasps are applied with more than one preshapes. At the bottom level, the preshape is also chosen per object shape and approach vector. Fig. 4 shows examples of objects that are included in the database.

Each grasping sequence in the database is generated by placing the hand in a specific configuration with respect to the object and then closing the fingers. For the recognition that relates to levels 1 and 2 in the recognition hierarchy (see Fig. 1), a simple spherical grasp strategy with a randomly chosen preshape is used. The spherical grasp strategy generates the approach direction for the hand by sampling the unit sphere around the center of mass of the object. Each sample then consists of a vector that points toward the center of mass of the object.

The strategy and the preshapes used for level 3 in the recognition hierarchy are shape specific. Therefore, strategies were developed for each shape used in the experiments. The hand preshapes for level 3 were generated with finger joint values in the interval $([-90; -70], [-10; 10])^{\circ}$, where the seventh joint was one of $90^{\circ}, 60^{\circ}$, and 0°, as shown in Fig. 5.

The following grasp strategies are applied for the shape primitives.

1) Sphere—The approach directions are sampled randomly from the unit sphere with origin in the center of gravity of the object. Both the ball preshape (60°) and the parallel preshape (0°) were used.
2) Cylinder—The object is approached either from the top or from the side. When approaching from the top, a ball grasp preshape is used, and the approach direction is pointing toward the object center of mass. For side grasps, the approach is sampled with an angle of 0–20° with respect to the horizontal plane, pointing toward the center of mass of the object. The preshape in the side grasp uses an angle of 0 on joint 7 so that a parallel grasp can be obtained.
3) Box—The object is approached using a vector lying in the plane defined by the world z-axis and the longest axis of the box and pointing toward the center of gravity. A parallel preshape of the hand is used.

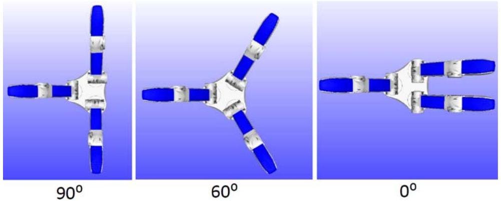


Fig. 5. Hand configuration when the seventh joint is at 90°, 60° and 0°.

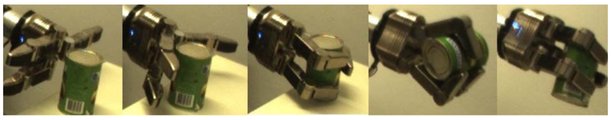


Fig. 6. Few examples from the execution of real experiments.

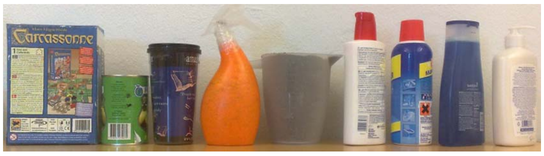


Fig. 7. Objects used in real experiments, with last three deformable.

In addition, two natural objects, i.e., the hamburger sauce and the bottle (see Fig. 4), used the same strategy as the cylinder. The tactile information and the joint configuration are recorded from simulation at regular time intervals.

In general, the performance of the simulation is largely dependent on the level of details of the geometries in both hand and objects. In our setup, to generate a simulated grasp using a modern quad core computer took approximately 2 s.

C. Generating Training Data on a Robot

The real-world experiments show the feasibility to assess grasp stability on physical robot platforms. The experiments aim to serve as a proof of concept rather than assessing the exact performance rates in different use cases. The experimental evaluation on real data follows the methodology used in simulation such that similar objects and same grasp types are used. The objects are placed such that they are initially not well centered with respect to the hand to assess the ability of the methods to cope with the uncertainty in pose estimation. A few example grasps are shown in Fig. 6. The real data include side grasps on the objects in Fig. 7 with the preshape shown in Fig. 5, where the seventh joint is 0°. After preshaping, the hand closes the fingers with equal speeds, while limiting the maximum torque

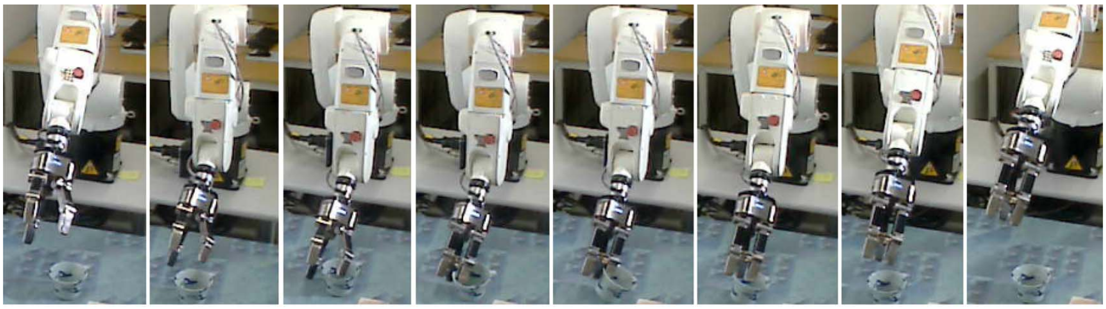


Fig. 8. Example of a failed grasp when only visual input is used. Details about the system are reported in [7].

of each actuator until reaching a static state where the object does not move or a fully closed hand configuration is reached. The latter occurs in the case of an unsuccessful grasp.

Tactile readings and corresponding joint configurations were recorded starting from the first contact until a static state is achieved. To generate stable/unstable label for a grasp, the object is lifted and rotated $[-120^\circ, +120^\circ]$ around the approach direction. The grasps where the object is dropped or moved in the hand were labeled as unstable. One hundred stable and 100 unstable grasps were generated for each object. Data processing, training, and classification followed the same methodology as described for the simulated data.

V. Experiments

We begin the experimental part by describing a simple demonstration scenario to show that the proposed approaches are viable in real applications. As the main experimental contribution, we then proceed to study the effect of different types of information for the estimation of grasp stability, starting with the experimental setup in Section V-C, followed by results in Sections V-D (one-shot recognition) and E (temporal recognition).

A. Demonstration

The feasibility of the approach is demonstrated in a realistic scenario. The demonstration is included to better show how the proposed methodology can be integrated in a real robotic system. Quantitative evaluation of the methodology is presented after the demonstration.

A vision-based system can provide information about the specific objects in the scene, and their pose [4]–[6] or potential grasping points on the object [7], [40]. In our previous work, we have shown how this can be done for known [4], unknown [5], [6], and familiar objects [7], [40]. However, in the previous work there were many cases that resulted in unsuccessful grasps. One example using system from [7] is shown in Fig. 8, and more examples are provided in the supplementary material.

The scenario that is demonstrated is as follows: Objects of the known geometry are placed in the workspace of a robot in a known position similar to [4]. Grasp hypotheses from a planner [41] are applied on the real robot by placing each of the five objects (see Fig. 9) in a known position. The planner is performing object decomposition for complex objects and plans grasps on the decomposed parts [4]. In our scenario, the planner is configured for a specific preshape. To demonstrate grasping of asymmetric objects in different poses, we place them in four different orientations with respect to the robot. After a suitable grasp is generated by the planner, the hand is moved to a preshape position, and the fingers are closed. After a steady state is reached (no change is detected in the tactile sensors), the stability of the grasp is estimated. Finger closing is controlled by executing a constant velocity motion for the finger joints and simultaneously limiting the maximum force by limiting the current for the finger actuators.

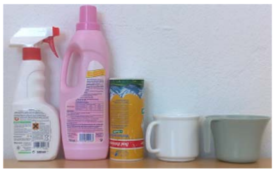


Fig. 9. Objects used to generate a dataset for the demonstration.

Algorithm 1 Calibration mode.

```
Choose a suitable grasping strategy for object $O$.
for $i = 1$ to $n$ do
    Preshape the hand
    Grasp object $O$ according to the chosen grasping strategy.
    Record tactile and joint configuration data during the grasp.
    Manipulate the object $O$ along a predetermined path.
    Record object motion relative to the hand $\Delta T$.
    if $\Delta T > 0$ then
        Grasp $i$ is unstable.
    else
        Grasp $i$ is stable.
    end if
end for
Using recorded data from each grasp $i$, train a classifier $C$.
```

Before the system can be operated, a training (calibration) process, which is required for each individual robotic hand, needs to be completed. The calibration process is described in Algorithm 1. The algorithm is run using the objects in Fig. 9; 114 stable and 114 unstable grasps are generated, including 58 grasps from the white spray bottle and 32 grasps from the pink detergent bottle in Fig. 9. While the calibration algorithm is not tied to a particular classification methodology,

Algorithm 2 Operation mode.

```
Generate a grasp using our grasp planner.
Preshape the hand.
Grasp object by closing fingers.
Evaluate classifier using sensor data.
if $P(S = stable) > P(S = unstable)$ then
    Lift object.
else
    Go to 1.
end if
```

in the demonstration, the HMM classifier that is presented in Section III-C is shown.

The operation mode of the demonstration system is described in Algorithm 2. A grasp is estimated as stable if the probability of a stable grasp exceeds the probability of the grasp being unstable, i.e., $P(S = \text{stable}) > P(S = \text{unstable})$. The probabilities are estimated using the well-known HMM "forward algorithm" to compute the probability of the observed sequence of measurements, assuming equal prior probabilities for stable and unstable.

Fig. 10 shows snapshot images from the operation of the system.[2] The robot attempts to grasp a bottle by first placing the hand in a preshape position given by the planner mentioned earlier, as shown in Fig. 10(a). Then, the fingers are closed as described earlier. The closed grasp is shown in Fig. 10(b) with the corresponding tactile measurements in Fig. 10(c). The grasp is predicted to be unstable, with the log-likelihood ratio $\log P(\text{unstable})/P(\text{stable})$ of the two models being $191.1270 > 0$, indicating unstable grasp. Now, in order to demonstrate that the failure was correctly predicted, instead of regrasping, the robot is nevertheless commanded to lift the object. The object drops as shown in Fig. 10(d), demonstrating the ability to correctly recognize an unsuccessful grasp. Next, to demonstrate that the stable grasps are also successfully recognized, another grasp that is generated by the same grasp planner is shown in Fig. 10(e). The closed grasp and the corresponding tactile measurements are shown in Fig. 10(f) and 10(g). Based on the measurements, the grasp is predicted to be stable, with the difference across log-likelihoods of the two models being $-537.7687 < 0$, indicating a stable grasp. Lifting and rotating the object around demonstrates this in Fig. 10(h), which concludes the demonstration.

B. Evaluation of Learning Capability

The experiments are divided according to the hierarchy presented in Section III. The goal is to evaluate the effect of the increasing knowledge on the classification results with both one-shot and temporal classification approaches.

1) Level 1 (No constraints): On this level, no constraints are placed on the data that are used for training the classifiers. In other words, only tactile sensor measurements and the joint configuration are available, and the other variables are unknown. The grasps are sampled from a sphere, and the hand is oriented toward the object. The data are collected in simulation across multiple object shapes and scales.

2) Level 2 (Constraints on object shape): The shape of the object is known, enabling the use of shape-specific classifiers. The grasps are randomly sampled from a sphere, and the hand is oriented toward the object. The data are collected in simulation.

3) Level 3 (Constraints on approach vector, preshape, and object shape): On level 3 of the hierarchy, constraints are placed on the approach vector, the grasp preshape, and the object shape. The data are collected using a manually chosen approach vector, and the preshape is adjusted to the shape of the object. On this level, the shape is known so that shape-specific classifiers can be used. Both simulated data and real data are available at this level.

C. Experimental Setup

1) Data: The simulated data that are used in the experiments consist of five objects with three different grasp configurations applied to them. Three of the objects have primitive shape (box, cylinder, and sphere), and two have natural shape (hamburger sauce and bottle). Each object is scaled to three different sizes: 0.75, 1.0, and 1.25 of the original size. For each object/size/grasp combination, 1000 unstable and 1000 stable grasps are randomly chosen from the database described in Section IV-B. Thus, each object/grasp dataset consists of 3000 stable and 3000 unstable grasps. When we refer to specific simulated object/grasp combination, terms *side* and *top* are used for grasps that are generated as side and top grasps, while *sph.* is used for grasps that are generated uniformly from a sphere around the object (random approach vector). Altogether, there are then 30 000 samples for the five objects. We also refer to the root node of the information hierarchy, which contains all samples of primitive shapes: a total of 18 000 samples.

The real data that are collected include nine objects with 100 unstable and 100 stable grasps for each object. Thus, there are 1800 samples in the real data set. The details of the real-data collection are described in Section IV-C.

2) One-shot recognition: As mentioned in Section III-B, we utilize the AdaBoost algorithm in one-shot classification. Because of the formulation of the AdaBoost, a weak learner needs to be chosen. In the experiments, a decision tree with a branching factor of 1 was used as the weak learner, effectively reducing the tree to a series of linear discriminants. The branching factor was determined from a series of tests that showed that using a branching factor of 1 performed as good as or better than larger branching factors on the data described in Section IV. Two hundred iterations of AdaBoost were run to find the final classifier in all experiments. For an SVM classifier, $\gamma = 0.03$, and constant C related to the penalty applied to incorrectly classified training samples [29] is set to $C = 0.4$.

All experiments are reported as tenfold cross-validation averages, except where otherwise noted. In each case, the datasets used to train and test the classifiers are balanced, i.e., the datasets contain equal number of unstable and stable grasps. Image moments are used as the feature representation for the one-shot

[2] See the supplementary video for a more detailed demonstration.

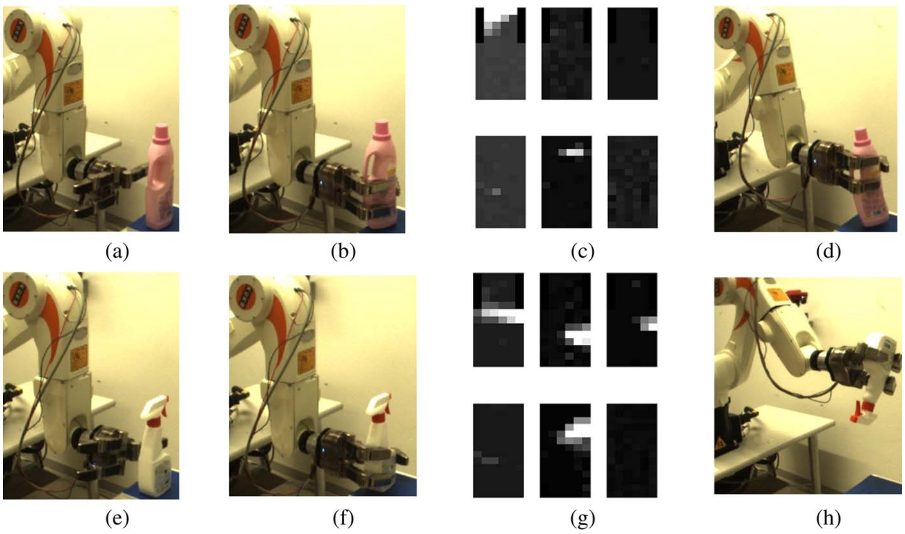


Fig. 10. Operation of the system. First row shows unsuccessful grasp, while the second row shows successful grasp. (a) and (e) Hand in a preshape position. (b) and (f) Closed grasp. (c) and (g) Tactile measurements. (d) Object dropped, while lifting. (h) Lifting and rotating the object successfully.

classifiers. The joint data in addition to the tactile data are also included in the features unless otherwise noted.

3) Temporal recognition: To study if the temporal information improves the recognition performance, two HMMs, one for stable grasps and another for unstable ones, were trained. The stopping criterion for HMM training was a convergence threshold of 10^{-4} with a 10-iteration limit. In order to improve the reliability of the evaluation, both ergodic and left-to-right HMM were evaluated independently. The reason for these multiple experiments is that by evaluating multiple temporal models, we aim to understand if the temporal ordering plays part in the modeling. The covariance of the mixture model component distributions was forced to be diagonal.

In the training of the temporal model, the structure of the HMM needs to be chosen in the form of structural parameters, which describe the number of HMM states and the number of mixture model components for each state. These were chosen experimentally such that the HMM was trained using different parameter settings, and the setting producing at least lowest equal error rate result (equal number of false positives and negatives) or better performance than that was chosen. The number of states was varied between 2 and 6, while the number of mixture components was between 2 and 5.

Experiments were performed both on simulated and real data. For simulated data, randomly chosen 80% of the samples were used for training and the other 20% for testing. For the real data, tenfold cross validation was used to evaluate the performance, and the best parameter setting over all folds was chosen.

Image moments were used as features, similar to one-shot learning. However, to reduce the number of parameters in HMM and speed up the training process, principal component analysis was applied to the moment and joint measurements separately to reduce the dimensionality of the dataset. The number of principal components was chosen such that at least 99% of the total variance is retained.

D. One-Shot Recognition Results

In this section, we present a collection of experiments based on the information hierarchy in Fig. 1 using the AdaBoost classifier. SVM classifier is used with image moments to examine the separability of the grasp stability at each level by means of log-likelihood histograms. We also study the effect of the joint configuration data on the classification by including or excluding them from the feature vector for the classifier when using real data. Training time for the classifiers is less than five min, for the reported amount of samples. Adaboost training time increases linearly with the amount of samples, while SVM training time increases quadratically. Classification of a single sample takes less than 10 ms with both of the presented classifiers. SVM classification time increases linearly with the amount of samples used for training.

1) Real data: The experiments begin by showing results using real data. Sampling grasps with a real hand is a slow process, and thus, the sample size is limited. To study the effect of the amount of samples used for training, we ran a series of tests with variable sample sizes. In each case, the same object was used both for training and testing. The results of these tests are shown in Table I, which shows the classification rates for training datasets of difference sizes. The test shows that for a specific grasp on the cylindrical object, 100 samples are already enough to reach classification performance levels achieved with higher amount of samples; the differences in classification performance above 100 samples are not statistically significant. However, this is the case only when the stable and unstable grasps are distinctive, i.e., we achieve a high rate of correctly classified grasps. In the case of the white bottle dataset, where the classification rate is lower, the results show that more than 200 samples could be useful in increasing the classification performance.

Classification results as percentages for single object classifiers (known object case) are presented in columns 2 and 3 of Table II. Classification rates are shown both with joint

TABLE I
ADABOOST CLASSIFICATION RATES (IN PERCENT) ON DATASETS WITH VARIABLE AMOUNT OF SAMPLES

Samples	50	100	150	200
Def. cylinder	74.6 %	85.0 %	84.8 %	89.0 %
W. Bottle	64.6 %	68.0 %	68.5 %	75.5 %

TABLE II
ADABOOST CLASSIFICATION RATES (IN PERCENT) ON KNOWN AND UNKNOWN OBJECTS WITH AND WITHOUT JOINT DATA

	Known obj.		Unknown obj.	
	w/j	wo/j	w/j	wo/j
Cylinder	88.9 %	90.3 %	80.4 %	81.9 %
Def. cylinder	91.0 %	89.0 %	76.0 %	76.5 %
Cone	79.5 %	81.0 %	73.0 %	68.0 %
O. Bottle	77.0 %	78.5 %	72.5 %	72.0 %
Shampoo	82.5 %	76.0 %	70.0 %	71.5 %
Pitcher	84.5 %	78.0 %	71.0 %	66.0 %
W. Bottle	76.0 %	73.5 %	75.0 %	76.0 %
B. Bottle	74.0 %	75.0 %	68.5 %	69.0 %
Box	89.0%	91.0 %	78.0 %	73.0 %

configuration data and without it, and the classification rates were computed for image moment feature representations. The main focus in this experiment is to study prediction of the grasp stability on known objects that the system has previously learned. The average classification rate for known objects is 82.5% including joint data and 81.4% excluding them from the measurements. Thus, the inclusion of joint data seems to benefit the recognition but only to a minor effect. Moreover, the result indicates that at least with known objects the proposed approach seems to have adequate recognition rate for practical usefulness.

We also study how well the trained system can cope with unknown objects, i.e., objects that have not been used to train the system. The results are shown as percentages of correct classification in columns 4 and 5 of Table II, which are adjacent to the results with known objects. The results are for a system that has been trained on all the objects except the object for which the classification rate is shown. The average recognition rate is 73.8% with joint data and 72.7% without them. The results show that while the classification rate is lower than with known objects, it is still possible to make predictions of the grasp stability on unknown objects to some extent. However, this holds true only when similar grasps are applied on unknown objects as were applied to the objects that the system were trained on. In comparison, including grasps from all objects, including the one being tested, for a single classifier yields a result of 78.6% correct classification across all the objects in the real object set. This indicates that the variety of objects that are used in training plays an important role in order to attain good performance and that the knowledge of object identity is useful but does not seem necessary if the training data include same or similar objects.

Two objects of a primitive shape are included in the real data: a box and a cylinder. Table III shows classification percentages when the classifier is trained only on one of the primitive objects. The classifier is then asked to classify the grasp stability of grasps made on real-world objects with different shapes. Cross validation was not needed in this case, because the training and test sets are naturally separate. The average classification rate for the cylinder model is 68.0% and for the box model 66.4%. These results no longer seem adequate for a real system, which again suggests that the variety in the training data is essential.

TABLE III
CLASSIFIER PERFORMANCE (IN PERCENT) WHEN TRAINING WITH A PRIMITIVE OBJECT

Trained object	**Cylinder**	**Box**
Def. cylinder	76.0 %	73.5 %
Cone	66.0 %	69.5 %
O. Bottle	64.5 %	61.0 %
Shampoo	66.5 %	64.0 %
Pitcher	71.0 %	62.0 %
W. Bottle	73.5 %	69.5 %
B. Bottle	58.5 %	65.0 %

TABLE IV
ADABOOST CLASSIFICATION RATES (IN PERCENT) ACCORDING TO THE INFORMATION HIERARCHY ON SIMULATED DATA

Level	**Node**	**Classification rate**
Level 1	Root	75.3 %
Level 2	Prim. cylinder sph.	73.5 %
	Prim. box sph.	79.2 %
	Prim. sphere sph.	77.0 %
Level 3	Prim. cylinder side	80.7 %
	Prim. cylinder top	67.6 %
	Prim. box side	83.5 %
	Prim. sphere side	78.5 %

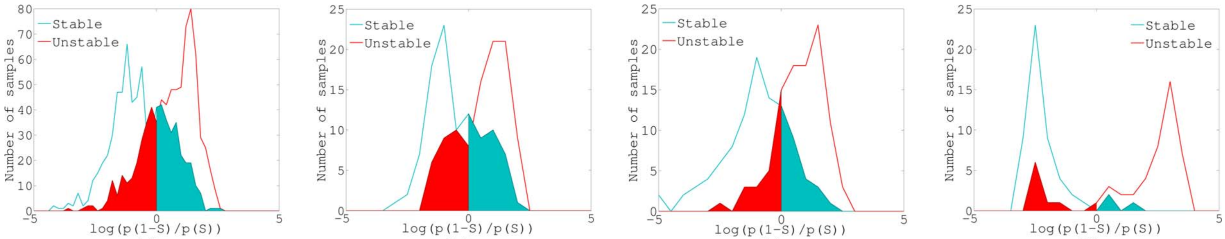
2) Simulated data: In contrast with the real data, in simulation we are able to sample a large number of grasps from different objects and using different grasp strategies. The following classification results were achieved using the simulated datasets described in Section IV. In Table IV, classification percentages are reported for each node in the information hierarchy. The root node (level 1) was randomly subsampled to 12 000 samples due to computational constraints and has classification rate of 75.3%. The average classification for level 2 (known object, unknown approach vector) is 76.5% and, for level 3 (known object, known grasp), 77.5%. A trend that increasing knowledge increases classification rate appears, similar to the experiments with real data. However, the trend is significantly weaker compared with the real data. Somewhat surprisingly, the real-data classification rates are notably higher when more information is available, and the trend is stronger, compared with simulation.

While the primitive shapes that are used in Table IV are simple shapes, we can use these primitive shapes to train the classifier and then use the classifier to classify grasps sampled from more natural, complex objects. The results are shown as percentages of correct classifications in Table V. Each row corresponds to a tested natural object (hamburger sauce and bottle), while each column corresponds to a combination of a training object and grasp strategy. Comparison results when training the classifier with the natural object and corresponding grasping strategy are shown in italic font. The figures in the table show that having data from the correct object has a notable positive effect on the classification rates. This is again a positive argument for the beneficial effect of a variety of training data.

Using the SVM and its ability to output estimates of the prediction certainty gives us a possibility to examine the performance of the classifier on different datasets in more detail compared with AdaBoost, which supports only the hard decision

TABLE V
AdaBoost Training With a Primitive Shape and Classifying Grasps Sampled from a Natural Object With Simulated Data

	Prim. cylinder sph.	**Prim. cylinder side**	**Prim. cylinder top**	**Prim. box sph.**	**Prim. box side**	**Prim. sphere sph.**	**Prim. sphere side**	**All classes sph.**
Hamb. sauce	71.5 % *78.7 %*	74.0 % *83.5 %*	62.9 % *72.4 %*	76.8 % *78.7 %*	73.6 % *82.0 %*	61.4 % *78.7 %*	62.7 % *83.5 %*	73.4 % *78.7 %*
Bottle	68.6 % *74.7 %*	77.4 % *82.0 %*	56.2 % *65.2 %*	72.6 % *74.7 %*	76.9 % *82.0 %*	59.4 % *74.7 %*	66.9 % *82.0 %*	69.7 % *74.7 %*



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

Fig. 11. Likelihood ratios for comparison of separability. (a) Root node, all objects, random grasp vector. (b) Cylinder, random grasp vector. (c) Cylinder side grasp. (d) Real cylinder side grasps.

TABLE VI
HMM Classification Rates (In Percent) on Datasets With Variable Amount of Samples

Object	50	100	150	200
Def. cylinder	86.7 %	85.0 %	85.4 %	87.0 %
W. Bottle	78.3 %	82.0 %	74.8 %	75.0 %

TABLE VII
HMM Classification Rates (In Percent) on Known and Unknown Objects

	LR, Kn.		ERG, Kn.		LR, Unkn.		ERG, Unkn.	
	w/j	wo/j	w/j	wo/j	w/j	wo/j	w/j	wo/j
Cyl.	90.0	86.5	92.5	82.0	83.0	77.5	81.0	75.0
Def. cyl	87.0	83.5	85.0	83.0	76.0	75.5	76.0	-
Cone	83.0	80.0	81.0	85.0	77.0	73.5	76.0	69.5
O. Bott.	74.0	76.5	75.0	73.5	77.5	77.5	74.5	77.5
Shamp.	81.0	77.5	78.5	77.5	81.0	75.5	79.0	75.0
Pitcher	83.0	81.5	84.0	73.5	72.5	77.5	72.5	65.0
W. Bott.	75.0	69.0	74.0	59.5	77.0	65.0	77.5	-
B. Bott.	78.5	71.0	75.0	66.0	75.5	69.0	75.0	-
Box	90.5	67.0	90.5	68.0	78.5	79.0	81.5	-

TABLE VIII
HMM Classification Rates (In Percent) When Training With a Primitive Object Only

Node	Cylinder		Box	
	LR	ERG	LR	ERG
Def. cylinder	67.0	69.5	74.0	74.5
Cone	66.0	66.0	70.0	76.5
O. Bottle	63.0	60.0	72.0	74.5
Shampoo	61.5	57.5	75.5	77.5
Pitcher	79.5	78.5	-	-
W. Bottle	58.5	50.0	76.5	76.5
B. Bottle	57.0	55.0	73.5	74.5

TABLE IX
HMM Classification Rates (In Percent) According to the Information Hierarchy on Simulated Data

Level	Node	LR	ERG
Level 1	Root	64.9	64.6
Level 2	Prim. cylinder sph.	70.2	70.2
	Prim. box sph.	62.1	59.0
	Prim. sphere sph.	77.4	76.9
Level 3	Prim. cylinder side	69.3	64.3
	Prim. cylinder top	69.5	69.3
	Prim. box side	68.6	69.0
	Prim. sphere side	62.8	63.2

boundary. This comparison can be seen in Fig. 11. In Fig. 11, log-likelihood ratios $\log 1 - P(S)/P(S)$ calculated from the probabilities for stable and unstable samples are shown in histogram form: red for unstable and light blue for stable. The classification errors are shown in filled color, with the filled area indicating the error probability. Fig. 11(a)–(c) is from simulated data, and Fig. 11(d) is from the real cylinder grasped with the SDH. It is evident from the figure that increasing information makes the distributions for stable and unstable grasps more separate, which was also indicated by the earlier results. Moreover, the figure also supports the finding that classifying the real data seems to be easier than the simulated data. Finally, the figure supports the use of probabilistic approaches for grasp classification, as the ability to measure the uncertainty in classification is important as it can, for example, allow tuning of the classification system to give fewer false positives.

E. Recognition Based on Temporal Model Results

In this section, we present HMM classification results that are obtained from the previously defined experiments. With given parameters, the training time for the HMM is less than 30 min for the reported amount of samples. The training time increases linearly with the amount of samples. Classification of a single sample takes less than 50 ms.

1) Real data: Similar to one-shot classification, we begin by investigating the general performance and the required number of samples to achieve good generalization properties. Table VI shows HMM classification percentages corresponding to Table I. The results demonstrate that the performance of the HMM classifier does not change much for distinctive grasps such as the ones from the deformable cylinder. While the average classification rates are similar to the one-shot model, the temporal

TABLE X
HMM TRAINING WITH A PRIMITIVE SHAPE AND CLASSIFYING GRASPS SAMPLED FROM A REAL-WORLD OBJECT WITH SIMULATED DATA

	cylinder sph.		**cylinder side**		**cylinder top**		**box sph.**		**box side**		**sphere sph.**		**sphere side**		**All sph.**	
	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG	LR	ERG
Hamb. sauce	61.2	60.8	63.3	60.3	57.8	57.3	59.2	57.3	63.1	61.1	51.6	52.9	65.2	63.2	59.3	59.6
	60.1	*57.2*	*67.5*	*68.0*	*68.1*	*64.8*	*60.1*	*57.2*	*67.5*	*68.0*	*60.1*	*57.2*	*67.5*	*68.0*	*60.1*	*57.2*
Bottle	58.4	58.3	67.6	64.3	63.1	65.4	58.4	54.2	57.2	-	52.4	52.8	62.7	59.6	57.4	58.5
	57.8	*55.6*	*65.8*	*66.8*	*68.8*	*69.1*	*57.8*	*55.6*	*65.8*	*66.8*	*57.8*	*55.6*	*65.8*	*66.8*	*57.8*	*55.6*

model seems to have better generalization capability in that the classification rate does not decrease significantly with smaller datasets.

Classification percentages for single object classifiers are presented in Table VII, both with joint configuration data (w/j) and without them (wo/j), to study the prediction capabilities on objects the system has previously learned with the two HMM types (left-to-right: LR, ergodic: ERG). The average classification rate for known objects (with joint data) is 82.4% with LR and 81.7% with ERG which are on a par with the one-shot learning (see Table II). Thus, with single object classifiers, the inclusion of temporal information did not increase classification performance.

Table VII also includes the results that study how well the trained system can cope with unknown objects, corresponding to Table II for the one-shot learning. The rates that are not included (marked with a dash) were below the level of chance. The results are similar in the way that the classification rates drop with unknown objects, the average rate with joint data being 77.5% for LR and 77.0% for ERG. However, the rate for unknown objects is in most cases high enough such that while the classification rate is lower than with known objects, it is still possible to make useful predictions of the grasp stability on unknown objects. LR seems to outperform ERG slightly in both cases, but the difference is not very significant. The reason for the difference is likely to be the simpler structure forced by the LR model, which in turn is likely to prevent overfitting. In comparison, using all data from all objects for a single classifier yields a result of 78.3% for LR model and 76.5% for ERG. It is remarkable that the difference between these and the results without the test object in the training data is less than 1%. Thus, with real data, it seems that the generalizability of grasp stability across objects is surprisingly good.

Table VIII shows classification results when the classifier is trained only on one of the primitive objects, corresponding to one-shot learning results shown in Table III. The average rate for cylinder primitive is 64.6% for LR and 62.3% for ERG, which are below the results of one-shot recognition. For box primitive, the recognition rate for pitcher was below level of chance and is thus not shown. On average, the rates for box primitive are nevertheless higher than those for the cylinder primitive and also higher compared with the one-shot learning. The cause of failure for the single object could not be identified. Altogether, the results are in agreement with those from one-shot learning in that the variety of training data seems important to attain good and stable performance.

2) Simulated data: Using the simulated data, Table IX reports the results for each node in the information hierarchy, corresponding to Table IV for the one-shot learning. For the LR model, the average classification for level 1 (root node, unknown object, and unknown approach vector) is 64.9%, 69.9% for level 2 (known object and unknown approach vector), and for level 3 (known object and known grasp), it is 67.5%. The results for ERG are similar. There are two observations to be made. First, these are consistently lower than those with one-shot learning, which is the opposite behavior compared with the real-data experiments, indicating that the simulated and real data do not match exactly. Second, the trend that increasing knowledge increases performance is broken for level 3, although the difference is not very significant. A possible explanation for this is that the stability of top and side grasps is on average more difficult to model with the HMM compared with modeling the stability of a grasp with random approach vector, because it is possible that some of the grasps with a random approach vector might be especially easy to recognize correctly.

The classification performance when training with primitive shapes but testing with real-world objects is shown in Table X, corresponding to Table V for the one-shot classification. The classification rates with the correct object are shown in italic font for comparison. The results indicate that on average the classification is significantly improved by having the correct object model instead of a general primitive model, again indicating the importance of variety in training data. Moreover, the results are again inferior to one-shot recognition, strengthening the finding that the temporal information is not essential for recognition with the available simulated data. To conclude, the real-world cases seem to contain dynamic phenomena that can be modeled better using a temporal model.

VI. CONCLUSION AND FUTURE WORK

Uncertainty is inherent to the activities that robots perform in unstructured environments. Probabilistic techniques have demonstrated the strength to cope with the uncertainty in robot planning, decision making, localization, and navigation. In the area of robot grasping, there have been very few examples to solve problems such as assessing grasp stability by taking uncertainty into consideration.

In this work, it was shown how grasp stability can be assessed based on uncertain sensory data using machine-learning techniques. Our learning framework takes into account object shape, approach vector, tactile data, and joint configuration of the hand. We have used a simulated environment to generate training sequences, including the simulation of the sensors. The methods were evaluated both on simulated and real data using a three-fingered robot hand. Our work demonstrates how grasp stability can be inferred using information from tactile sensors, while grasping an object before the object is further manipulated or during the manipulation step. We have implemented and evaluated both one-shot and temporal learning techniques. The temporal information was found to somewhat increase generalization capabilities in that a smaller number of training examples were needed and that the generalization performance to new objects was slightly increased. These come with the cost of increased computational complexity. One focus of the experiments was to study prediction capabilities of the proposed methods for known objects. We have also studied how the system can cope with unknown objects, i.e., objects that have not been used in the training step. The results show that while the classification rate is lower than with known objects, it is still possible to make useful predictions of the grasp stability on unknown objects. In summary, the experimental results show that tactile measurements allow assessment of grasp stability. The aim of this paper was not a perfect discrimination between successful and unsuccessful grasps but, rather, a measure of certainty of grasp stability. This also means that a system may be built to reject some stable grasps while having fewer unstable grasps classified as stable ones. Experiments showed that using sequential data to evaluate grasp stability appears to be beneficial during dynamic grasp execution.

Our current study proceeds in several directions. First, we are in the process of integrating the presented system with a vision-based pose estimation system and grasp planning. Second, we are implementing a grasping system based on the proposed ideas for local control of grasps and corrective movements. In both cases, the aim is to demonstrate a robust object grasping and manipulation system for both known and unknown objects based on visual and tactile sensing. Finally, we have developed a more elaborated probabilistic framework in which we study the joint probability of object-relative gripper configurations, tactile perceptions, and grasping feasibility. Here, we have developed a kernel-logistic-regression model of pose- and touch-conditional grasp success probability. The goal is to show how a learning framework can be used for grasp transfer, i.e., if the robot has learned how to grasp one type or category of objects, to use this knowledge to grasp a new object.

References

[1] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin, Germany: Springer, 2008, ch. 28, pp. 671–700.

[2] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[3] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Krueger, O. Kroemer, J. Peters, and J. Piater, "Learning continuous grasp affordances by sensorimotor exploration," in *From Motor Learning Interaction Learning Robots*, 1st ed., O. Sigaud and J. Peters, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 451–465.

[4] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping known objects with humanoid robots: A box-based approach," presented at the 14th Int. Conf. Adv. Robot., Munich, Germany, Jun. 2009.

[5] B. Rasolzadeh, M. Bjorkman, K. Huebner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in real world," *Int. J. Robot. Res.*, vol. 29, nos. 2–3, pp. 133–154, 2010.

[6] M. Popovic, D. Kraft, L. Bodenhagen, E. Baseski, N. Pugeault, D. Kragic, T. Asfour, and N. Kruger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robot. Auton. Syst.*, vol. 58, no. 5, pp. 551–565, 2010.

[7] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robot. Auton. Syst.*, vol. 58, no. 4, pp. 362–377, 2010.

[8] M. Shimojo, T. Araki, A. Ming, and M. Ishikawa, "A high-speed mesh of tactile sensors fitting arbitrary surfaces," *IEEE Sensors J.*, vol. 10, no. 4, pp. 822–830, Apr. 2010.

[9] M. Higashimori, M. Kaneko, A. Namiki, and M. Ishikawa, "Design of the 100 g capturing robot based on dynamic preshaping," *Int. J. Robot. Res.*, vol. 24, no. 9, pp. 743–753, 2005.

[10] H. Wakamatsu, S. Hirai, and K. Iwata, "Static analysis of deformable object grasping based on bounded force closure," in *Proc. Int. Conf. Robot. Autom.*, Minneapolis, MN, Apr. 1996, pp. 3324–3329.

[11] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1824–1829.

[12] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4679–4684.

[13] A. M. Howard and G. A. Bekey, "Intelligent learning for deformable object manipulation," *Auton. Robot.*, vol. 9, no. 1, pp. 51–58, 2000.

[14] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," presented at the Rob.: Sci. Syst., Workshop Robot Manipulation: Sensing Adapting Real World, Atlanta, GA, 2007.

[15] M. Prats, P. Sanz, and A. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3975–3980.

[16] S. Ekvall and D. Kragic, "Learning and evaluation of the approach vector for automatic grasp generation and planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4715–4720.

[17] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Bayesian estimation for autonomous object manipulation based on tactile sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 707–714.

[18] A. Jiméneza, A. Soembagijob, D. Reynaertsb, H. V. Brusselb, R. Ceresa, and J. Ponsa., "Featureless classification of tactile contacts in a gripper using neural networks," *Sens. Actuators A: Phys.*, vol. 62, nos. 1–3, pp. 488–491, 1997.

[19] S. Chitta, M. Piccoli, and J. Sturm, "Tactile object class and internal state recognition for mobile manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 2010, pp. 2342–2348.

[20] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," presented at the 14th Int. Conf. Adv. Robot., Munich, Germany, Jun. 2009.

[21] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, St. Louis, MO, 2009, pp. 243–248.

[22] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, "Haptic object recognition using passive joints and haptic key features," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 2010, pp. 2349–2355.

[23] K. Hsiao, L. Kaelbling, and T. Lozano-Perez, "Task-driven tactile exploration," presented at the Robot.: Sci. Syst. Workshop Representations Object Grasping Manipulation in Single Dual Arm Tasks, Zaragoza, Spain, Jun. 2010.

[24] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, and V. Kyrki, "Learning grasp stability based on haptic data," presented at the Robot.: Sci. Syst. Workshop Representations Object Grasping Manipulation in Single Dual Arm Tasks, Zaragoza, Spain, Jun. 2010.

[25] "Weiss robotics tactile sensor," (Apr. 22, 2011). [Online]. Available: http://www.weiss-robotics.de/en.html

[26] J. Laaksonen, V. Kyrki, and D. Kragic, "Evaluation of feature representation and machine learning methods in grasp stability learning," in *Proc. IEEE-RAS 10th Int. Conf. Humanoid Robot.*, 2010, pp. 112–117.

[27] Y. Freund and R. E. Shapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learning*, 1996, pp. 148–156.

[28] A. Vezhnevets. (2006). *GML AdaBoost Matlab Toolbox*, [Online]. Available: http://graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox.

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

[30] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[31] C.-C. Chang and C.-J. Lin. (2001). *LIBSVM: A Library for Support Vector Machines*. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[32] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in Large Margin Classifiers*, Cambridge, MA: MIT Press, 1999, pp. 61–74.

[33] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[34] K. Murphy. (2005). *Hidden Markov Model (HMM) Toolbox for Matlab*. [Online]. Available: http://www.cs.ubc.ca/murphyk/Software/HMM/hmm.html

[35] J. Jorgensen, L. Ellekilde, and H. Petersen, “RobWorkSim—An open simulator for sensor based grasping,” presented at the 41st Int. Symp. Robot. ROBOTIK, Munich, Germany, Jun. 2010.

[36] A. T. Miller and P. K. Allen, “Graspit! A Versatile Simulator for Robotic Grasping,” *IEEE Robot. Autom. Mag.*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[37] J. A. Jorgensen and H. G. Petersen, “Usage of simulations to plan stable grasping of unknown objects with a 3-fingered Schunk hand,” presented at the IROS Workshop Robot Simulators, Nice, France, Sep. 2008.

[38] C. Ferrari and J. Canny, “Planning optimal grasps,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Nice, France, May 1992, pp. 2290–2295.

[39] R. Suárez, M. Roa, and J. Cornella, “Grasp quality measures,” Tech. Univ. Catalonia, Barcelona, Spain, Tech. Rep. IOC-DT-P-2006-10, 2006.

[40] J. Bohg and D. Kragic, “Grasping familiar objects using shape context,” presented at the 14th Int. Conf. Adv. Robot., Munich, Germany, Jun. 2009.

[41] K. Huebner, “BADGr: A toolbox for box-based approximation, decomposition and grasping,” presented at the IROS Workshop Grasp Planning Task Learning Imitation, Taipei, Taiwan, Oct. 2010.

Yasemin Bekiroglu received the M.Sc. degree in computer engineering from Karadeniz Technical University, Trabzon, Turkey, and the M.Sc. degree in applied artificial intelligence from Dalarna University, Dalarna, Sweden.

Since 2008, she has been working toward the Ph.D. degree with the Department of Computer Science, Royal Institute of Technology, Stockholm, Sweden, in the field of robotic grasping.

Janne Laaksonen received the M.Sc. degree in information technology from the Lappeenranta University of Technology (LUT), Lappeenranta, Finland, in 2008. He is currently working toward the Ph.D. degree with the Machine Vision and Pattern Recognition Research Group, Department of Information Technology, LUT.

His current research interests include sensor-based robotic grasping and manipulation under uncertainty.

Jimmy Alison Jørgensen received the M.Sc. and Ph.D. degrees in computer systems engineering from the University of Southern Denmark (SDU), Odense, Denmark, in 2007 and 2010, respectively.

He is currently a Postdoctoral Fellow with the Maersk-McKinney Moller Institute, SDU. He is one of the main developers of the open-source robotics framework RobWork and is responsible for the RobWorkSim simulation package. His main research interests include robotics, dexterous grasping, and applied robotic simulation.

Ville Kyrki (M’04) received the M.Sc. and Ph.D. degrees in computer science from the Lappeenranta University of Technology (LUT), Lappeenranta, Finland, in 1999 and 2002, respectively.

He is currently a Professor of computer science with the Department of Information Processing, LUT, serving as the Head of the Machine Vision and Pattern Recognition Laboratory. His research interests include real-time vision for robotics, multi-sensor estimation and control, and robot learning.

Dr. Kyrki is the Co-Chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Computer and Robot Vision. He is a member of the IEEE RAS and the International Association of Pattern Recognition.

Danica Kragic (M’09) received the M.Sc. degree in mechanical engineering from the University of Rijeka, Croatia, in 1995 and the Ph.D. degree in computer science from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2001.

She is a Professor of computer science with the School of Computer Science and Communication and the Acting Director of the Centre for Autonomous Systems. Her research interests include grasping, visual servoing, active vision, and activity recognition.

Dr. Kragic received the IEEE Robotics and Automation Society Early Academic Career Award in 2007. In 2008, she received the Swedish Foundation for Strategic Research Future Research Leaders Award. Since 2006, she has been the Chair of the IEEE Robotics and Automation Society (RAS) Technical Committee on Computers and the Robot Vision and a member of the IEEE RAS Conference Editorial Board.

2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Integrating Grasp Planning with Online Stability Assessment using Tactile Sensing

Yasemin Bekiroglu, Kai Huebner and Danica Kragic

Abstract— **This paper presents an integration of grasp planning and online grasp stability assessment based on tactile data. We show how the uncertainty in grasp execution posterior to grasp planning can be dealt with using tactile sensing and machine learning techniques. The majority of the state-of-the-art grasp planners demonstrate impressive results in simulation. However, these results are mostly based on perfect scene/object knowledge allowing for analytical measures to be employed. It is questionable how well these measures can be used in realistic scenarios where the information about the object and robot hand may be incomplete and/or uncertain. Thus, tactile and force-torque sensory information is necessary for successful online grasp stability assessment. We show how a grasp planner can be integrated with a probabilistic technique for grasp stability assessment in order to improve the hypotheses about suitable grasps on different types of objects. Experimental evaluation with a three-fingered robot hand equipped with tactile array sensors shows the feasibility and strength of the integrated approach.**

I. INTRODUCTION

Grasping is an essential skill for a general purpose service robot to work in an industrial or home-like environment. A successful grasp is often described as a relationship between an object and a gripper that allows for further manipulation of the object. Given that some of the object and gripper parameters such as pose, shape, weight and/or material properties are known, grasp planning can be performed. For this purpose, analytical approaches have been developed and improved over the last two decades, as [1], [2], [3], and recently summarized in [4]. These achievements enabled simulation of grasps and evaluation of grasp planners on a measurable basis. Thus, most state-of-the-art simulation environments for grasping, e.g., GraspIt! [5], RobWorkSim [6] or OpenGrasp [7], employ analytical measures to compute grasp stability in simulated scenarios. The possibility of performing extensive and efficient experiments in simulation alleviates the maintenance of expensive equipment. It also provides information about the necessary parameters for stability analysis, like contact surfaces, center of mass, or friction coefficients.

In unstructured real-world environments, however, these parameters are uncertain, which presents a great challenge to the aforementioned approaches. Extraction and modeling of appropriate sensor data is commonly used to alleviate the problem of uncertainty. For instance, visual input is commonly used to determine the pose or shape of an object [8]. However, the accuracy of vision in terms of object pose estimation is limited even for known objects, as small errors in object pose are common and may cause failures in grasping. In terms of object shape approximation, algorithms are far from approximating objects at an accuracy that makes the necessary parameters comparable to their real values.

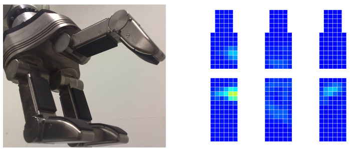


Fig. 1. The Schunk Dexterous Hand and example tactile array readings.

The consequences of such failures at the planning stage are difficult to prevent at the grasp execution stage, even though tactile and finger force sensors can be used to reduce them. Still, a grasp may fail even when all fingers have adequate contact forces and the hand pose is not much different from the planned one, e.g. if slippage or collision occur, [9].

We can therefore conclude that in realistic applications, grasp stability can not be assessed without the possibility of an error. In addition, the relationships between grasp stability, available sensory information, and actuator accuracy are embodiment-specific and inherently complex. To cope with these issues, we propose a probabilistic approach to model the relationship between tactile sensory information (see Fig. 1) and grasp stability. This strategy also allows for detection of failures at the grasp execution stage, such that objects can be regrasped before attempting to further manipulate them.

The contributions of our work are identified together with the related work in Section II. Section III presents the data generation using a grasp planner and Section IV stability assessment using Hidden Markov Models (HMMs). In Section V, the experimental setup and the experimental evaluation are presented. We conclude our work in Section VI.

II. RELATED WORK

In robotic object grasping there has been a lot of effort during the past few decades, see [8] for a recent survey. Most of the state-of-the-art grasp planning approaches first model the object shape with a number of primitives such as boxes [10], cylinders, cones, spheres [11], or superquadrics [12] to reduce the space of possible grasps. The decision about a suitable grasp is then made based upon analytical grasp quality measures. Such analytical approaches rely on accurate knowledge about the contacts between the hand

This work was supported by EU through the project CogX, IST-FP7-IP-215181 and Swedish Foundation for Strategic Research. The authors are with the Computer Vision and Active Perception Lab, Centre for Autonomous Systems, School of Computer Science and Communication, KTH, Stockholm, Sweden. {yaseminb,khubner,dani}@kth.se.

and the object in order to estimate the stability of a grasp. Therefore, few of these planners have been applied on real robot platforms, but in simulation only.

To operate in scenarios where exact knowledge of the object and the hand is not available, tactile and/or force-torque sensors can be used for control once in contact with the object. For instance, [13] proposes to maximize the tactile contact surface for removing a book from a bookshelf. In [14] force, visual and tactile feedback are combined to open a sliding door similarly to our previous work [15], [16].

In [17], the 3D pose of an object is determined based on tactile information. Similar work is presented in [18], where object localization is performed with knowledge of tactile contacts on specific objects. The surface type (e.g., edge, flat, cylindrical, sphere) of the tactile contact is determined using a neural network in [19]. In [20], tactile information is extracted from the sensors on a two-fingered gripper to distinguish properties of objects such as the open/closed and fill state.

There has also been work on object shape learning based on tactile data. The approaches use either one-shot data from the final state of the grasps [21], [22] or temporal data collected throughout the grasp or manipulation execution [20], [23]. In [21], a bag-of-words approach is presented which aims to identify objects using touch sensors available on a two-fingered gripper. The approach processes tactile images collected by grasping objects at different heights. In [22], a similar approach is taken for a humanoid hand. A more traditional approach to learning is employed with features extracted from tactile images in conjunction with hand joint configurations as input data for the object classifier. In [23], a measure of entropy is used to identify the most useful features for object recognition. In this case, a plate covered with tactile sensor is used as the manipulator.

In summary, the focus in the above is either the use of tactile data for object manipulation control or exploration of object properties, but not the stability assessment. Learning or analyzing object properties only through tactile sensors does not answer the question of grasp stability. In our earlier work, [24], [25], we have presented initial results of grasp stability prediction using tactile sensors, where part of the evaluation was performed in simulation. In this paper, we extend our work along the following lines: (i) We concentrate on grasp stability implementation and evaluation on a real system rather than in simulation. (ii) For this purpose, we integrate our method with a simulation-based grasp planner, in order to (iii) analyze the relation between analytical stability in simulation and real grasps under consideration of a range of uncertainties. (iv) For our learning mechanism, we apply an extended evaluation strategy based on the following stability degree formulations. We label a grasp as *reached* if it was at least possible to move the hand into the planned pose. If *lifted*, the object was also successfully lifted up vertically. *Transported* means a stable transport to a fixed location without dropping. The two last stages describe if the object was successfully *rotated* in the transported location and finally if the grip was so firm that it could be manually *pushed*. We mark a grasp as stable when it at least reached *rotated* state.

III. DATA ACQUISITION AND GRASP PLANNING

In our previous work [25], we generated the training data by applying side and top grasps on a set of training objects. The position of the object was changed slightly with respect to the robot hand over a number of test trials. In order to acquire two training sets of tactile readings – one to classify stable grasps, and one for unstable ones – we manually annotated grasps as stable / unstable. As a major assumption, grasps were aimed at the center of mass of each object.

In this paper, we automate the grasp planning and stability labeling by using a part-based grasp planner. Grasps on parts are more useful in terms of task-oriented grasping, and they represent a good trade-off between learning from random grasps and learning from manual side- and top-grasps only. Specifically, using a grasp planner that prunes the space of possible grasps, and thereby also tactile patterns, simplifies the classification as well as acquisition of training data on a real platform. On the other hand, part-based grasping is more challenging than our earlier restriction to grasps centered

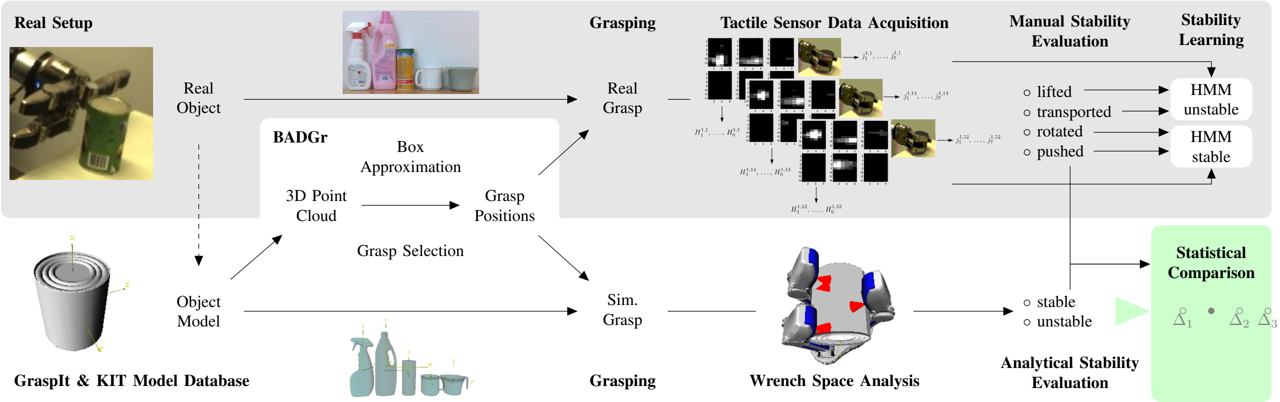


Fig. 2. We generate grasp positions for 5 known objects using the BADGr framework. We compare acquired stability levels of both the real (top) and the simulated (bottom) execution of each grasp. For the real setup, we acquire sensor data to learn real stability and unstability from two HMMs. The light green box on the right bottom represents a side view of a box for a part of an object. The offset distances mentioned in Sec. III-A and Tab. I are described by Δ_i.

on the center of mass. Especially for part-based grasps, the center of mass is an important factor for stability. For the experimental evaluation, we selected a publicly available box-based grasp planner, BADGr [26]. We illustrate the complete processing pipeline of our data acquisition in Fig. 2.

The grasp planner allows us to approximate given objects with primitive box shapes based on an efficient minimum volume bounding box implementation. Except for the work presented in [27], this grasp planner has not been evaluated on a real robot. In our case, we use 'known objects' in a similar way by first generating object-centered grasps offline. In our setup, the pose of each object will be fixed. This allows us to evaluate and compare and their analytical stability in GraspIt [5] with our evaluation of stability in the real world.

A. Results for Analytic Grasp Analysis in Simulation

In Tab. I, we present the results generated for 5 test objects, and divide into those detected as stable or unstable through analytic analysis in the simulator. For the asymmetric objects, a range over all 4 orientations is shown. For our later training, we aim at approximately 20 samples per object and stability class. We therefore increase the number of grasps by adding 3 offset distances that influence the palm position along the approach vector. Given a side-view of a box around a part, (see the light green box in Fig. 2), and an approach vector (see the arrow from its left), these Δ_i decentralize the original grasp center position on the object.

B. Results for Real Grasp Analysis

The planned grasp hypotheses were applied on the real robot by placing each of the 5 objects (Fig. 2) in a known position. The labels obtained by executing the simulated grasps on the real setup are given in Tab. I. The column *planned grasps* and $+3\Delta$ *distances* include the ranges for the number of hypotheses produced by the grasp planner depending on the orientations of the objects and the corresponding ranges extended by the offset distances. For example, for the *green cup*, 2 out of 8 unstable and 1 out of 21 stable grasps in simulation, are labeled as stable (being *rotated*) and unstable (being *reached*) respectively in experiments on the robot. In order to have equal number of stable and unstable grasps, some grasps are repeated as seen in the last column of the table. For a small and light object such as the *cup*, predictions in simulation are in general correct. For the cylindrical *salt bottle*, which is relatively small but heavier than the cup, predictions for unstable grasps are less reliable with the highest rate of unstable grasps in simulation labeled as stable in real experiments, (46.66%). Predictions about stable grasps for this object are in line with the real experiments. The highest rate of stable grasps which are labeled as unstable in real experiments (38.46%) were obtained from the deformable *pink bottle*. The most complicated object in shape, the *spraybottle*, provided relatively high contradictory results, 34.61% unstable grasps labeled as stable and 30% stable grasps labeled as unstable in the real experiments.

TABLE I

STATISTICS ON STABILITY IN SIMULATION AND REAL EXPERIMENTS.

	In Simulation			Real Evaluation					Dataset	
Object	labels	planned grasps	$+3\Delta$ distances	reached	lifted	transported	rotated	pushed	stable	unstable $_{\text{+repeated}}$
Cup (white)	s	6-9	27-36	*2*	-	-	**8**	**10**	**20**	19_{+1}
	u	3-11	12-48	*17*	-	-	**2**	-		
Cup (green)	s	9-10	21-26	*1*	-	-	**1**	**19**	**22**	7_{+15}
	u	3-6	24-38	*5*	*1*	-	**2**	-		
Salt (round)	s	11	40	*5*	*1*	-	**7**	**14**	**28**	14_{+14}
	u	5	24	*5*	*1*	*2*	**6**	**1**		
Bottle (pink)	s	5-8	17-19	-	*3*	*7*	**7**	**9**	**18**	*18*
	u	2-14	14-62	*5*	*3*	-	**2**	-		
Bottle (spray)	s	5-6	15-23	*4*	*2*	*3*	**7**	**14**	**30**	26_{+4}
	u	11-16	45-65	*2*	*7*	*8*	**8**	**1**		

From the table, we note that grasps that are supposed to succeed may fail due to several reasons which will be discussed shortly. Two example grasps are given in Fig. 3 to demonstrate how grasps may be labeled differently from the planned ones after execution on the real setup. As seen in Fig. 3, the grasps during the real experiments fail, although they were generated as stable grasps in simulation. Five different grasps (Fig. 4) were chosen and repeated 10 times to see the variance in the labels assigned by the real evaluation, Tab. II. While in one case, G_1, a stable grasp can be unstable in all trials, other stable grasp examples in simulation, G_4, G_5, could be stable in real experiments or may result in different levels of unstability in some trials such as grasp G_3. An unstable example, G_2, can also be stable in some trials. In summary, these experiments show that even the same grasps may be classified differently in real experiments, which therefore motivates the need for online assessment.

C. Analytical vs. Real Grasps

The above analysis exemplifies typical uncertainties and variances emerging for several reasons: not only state-of-the-art simulation environments lack modeling real correspondences, but also repetition of the same grasps may result in variable outcomes. We sketch the major issues immanent in our comparison of GraspIt and our real system:

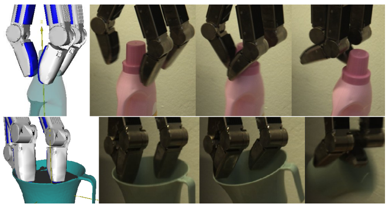


Fig. 3. Grasps that are stable in simulation but failed in real experiments.

TABLE II

VARIANCE OF LEVELS OF STABILITY FOR 5 GRASPS. GRASP NAMES ARE GIVEN WITH (GRASPIT LABEL, REAL LABEL).

Grasp	Reached	Lifted	Transp.	Rotated	Pushed
G_1(S, Reached)	10	-	-	-	-
G_2(U, Lifted)	2	1	4	3	-
G_3(S, Transp.)	3	3	-	4	-
G_4(S, Rotated)	-	-	-	5	5
G_5(S, Pushed)	-	-	-	6	4

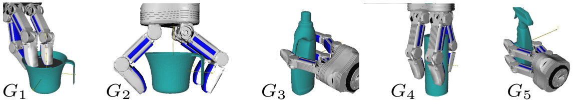


Fig. 4. Grasps used in the repeated experiments.

1) Perception: In our study, we use static poses for known objects. However, even the human inaccuracy in placing an object with the same pose is a factor that can make repetition difficult. In automated machine vision scenarios, e.g. pose estimation of known, or surface estimation of unknown objects, we can assume such accuracy to be even worse.

2) Actuator Control: Similarly to a slight variance in the object's pose, inaccuracies in the joint positioning of both robot arm and hand, or in dynamic grasp control may have strong effects on the success of a grasp.

3) Contact & Sensor Models: For the sake of efficiency, contact models, e.g. point contacts or soft contacts, may be inaccurately or only partially provided in a simulated environment. In a similar way, simulated models of the tactile sensors have to be reliable to match the real world.

4) Physics & Dynamics: Another key to bring simulation closer to the real world are robust physics and dynamics engines. As in the real world, an object should be dynamically affected by any forces, e.g. exerted by the fingers or gravity.

5) Object Properties: Linked to the dynamics of the scene, knowledge about various object properties is fundamental. On a basic level, this is often approached by mapping discrete material properties, e.g. plastic, metal or glass, to friction cone angles that define the grasp wrench space. Other properties, as deformability, center of mass, or wearout of objects and hands, are much more difficult to model.

These considerations emphasize the strengths of a method for online stability assessment in the real world, using learning techniques and close loop methods. Our approach to this problem by using Hidden Markov Models (HMMs) will be formalized in the next section.

IV. ONLINE STABILITY ASSESSMENT USING HIDDEN MARKOV MODELS

A. Feature Representation

In this work, we work with a three-fingered Schunk Dexterous Hand (SDH) with seven degrees of freedom and six two-dimensional tactile patches (see Fig. 1). Tactile measurements and corresponding joint configurations are recorded starting from the first contact with the object until a steady-state is reached. The data can be described by the following notation. An observation sequence with T observations is denoted by $x_1, .., x_T$. An observation at time instant t is denoted by $x_t = [M_f^t j_v^t]$, where $f = 1, .., F$, with F the number of tactile sensors, and $v = 1, .., V$, with V the number of joints of the robot hand. M_f^t includes features extracted from the tactile readings H_f^t on the sensor f at time instant t and j_v^t is a joint angle at time instant t.

For the SDH, we store 3×(14×6) readings on proximal and 3×(13×6) on distal sensors, plus 7 parameters representing the shape of the hand given the joint angles. Example images from the sensors are shown at the Tactile Sensor Data Acquisition stage in Fig. 2 during a stable grasp of a cylinder. Considering the two-dimensional tactile patches as images, we employ image moments $m_{p,q}$ as a representation in order to reduce the dimensionality. Moments are given by

$$m_{p,q} = \sum_z \sum_y z^p y^q H(z, y), \qquad (1)$$

where p and q represent the order of m, z and y the horizontal and vertical position on the tactile patch, and $H(z, y)$ the measured contact. We compute moments up to order two, $(p + q) \in \{0, 1, 2\}$, for each sensor array separately, which then correspond to the total pressure and its distribution in the horizontal and vertical direction. Therefore, six features for each sensor result in an observation $x_t \in \mathbb{R}^{6F+V}$. Finally, moment features and finger joint angles are normalized to zero-mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

B. Hidden Markov Models (HMMs)

Time-series grasp stability assessment is performed using Hidden Markov models [28]. In this section, we provide a basic description for the HMM-based inference. More details can be found in [25]. As sketched in Fig. 2, we train two HMMs: one representing stable and one unstable grasps. Classification of a new grasp sequence is performed by evaluating the likelihood of both models and choosing the one based on Maximum Likelihood approach. In this work, we evaluate both ergodic (fully connected) and left-to-right HMMs. The estimation of the HMM model parameters is based on the Baum-Welch procedure. The output probability distributions are modeled using Gaussian Mixture Models (GMMs). Unknown parameters are estimated from the training sequences $o = (x_1, .., x_T)$.

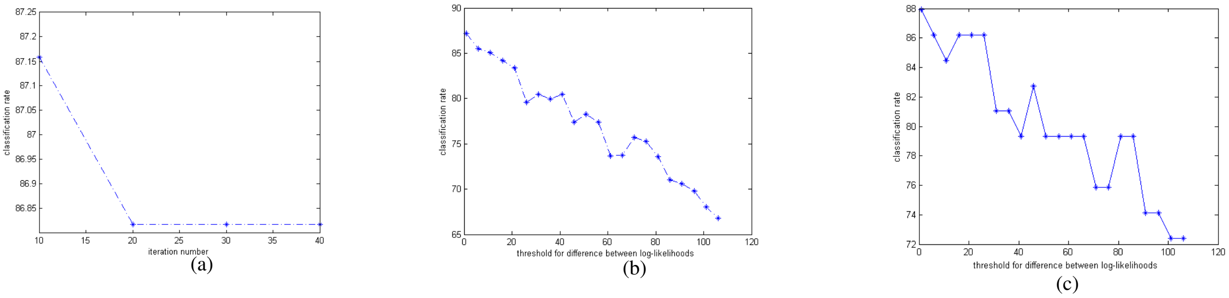
C. Principal Component Analysis

Principal Component Analysis (PCA) is used to reduce the dimensionality of the dataset which in turn reduces the number of HMM parameters and speeds up the training. PCA determines the directions along which the variability of the data is maximal. Let $\vec{M}^t = (M_1^t, .., M_F^t)$ be the features extracted from tactile sensor readings and $\vec{j}^t = (j_1^t, .., j_V^t)$ be the corresponding joint configurations. We apply PCA separately to both sets of variables $\vec{M}^t$ and $\vec{j}^t$ and project the dataset onto their respective basis of Eigenvectors. Principal components are extracted from the final observations in the training sets in order to reduce the computations. The principal components describing 99% of the total variance are used.

V. EXPERIMENTAL EVALUATION

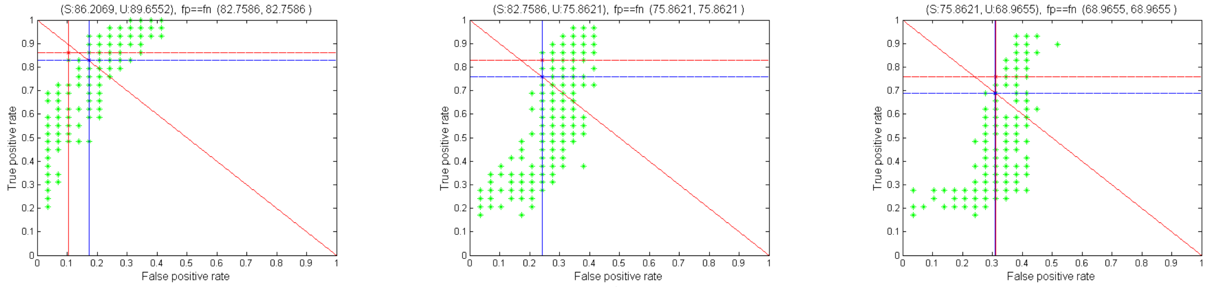
A. Experimental Setup

To examine the recognition performance, two HMMs, one for stable grasps and another for unstable ones were trained with the stopping criteria being the convergence threshold 10^{-4} with an iteration limit. Different numbers of iterations were applied as seen in Fig. 5(a). Since increasing the iteration number did not improve performances, an iteration limit of 10 was chosen for the experiments. In order to improve the reliability of the evaluation, both ergodic



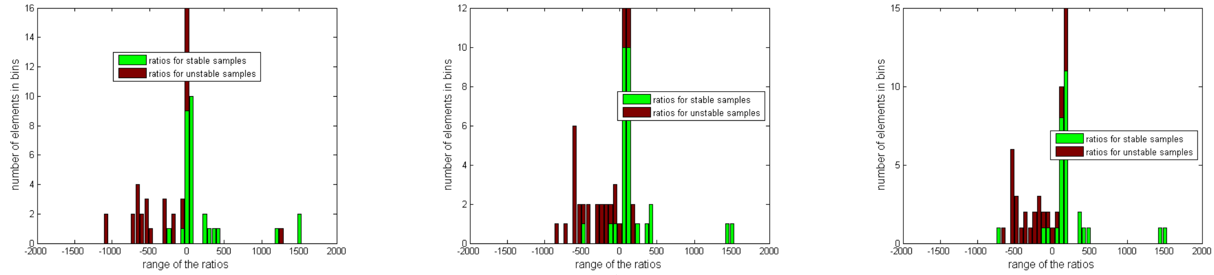
(a) (b) (c)

Fig. 5. Classification rates with different (a) iteration numbers using cross validation, and thresholds with (b) and without (c) cross validation.



(a) (b) (c)

Fig. 6. HMM model selection: (a) with th_1; (b) with th_2; (c) with th_3.



(a) (b) (c)

Fig. 7. The distribution of log-likelihood ratios: (a) with th_1; (b) with th_2; (c) with th_3.

and left-to-right HMM were evaluated independently with different structure parameters. In general, ergodic and left-to-right HMMs had comparable results, hence the experimental results were given with left-to-right structure. The range of 2-6 for the number of states and 2-5 for the number of components in a mixture were evaluated. The covariance matrix structures were forced to be diagonal. From these settings, two types of experimental results are provided with the aim of demonstrating the performance in detail: with and without cross validation. The experimental results with 10-fold cross validation are presented with the best parameters over all folds. For the experiments in which we do not apply cross validation, the data is separated randomly into a training (75%) and testing set (25%). With given parameters, the training time for the HMM is a few minutes for the described dataset linearly increasing with the iteration number. Classification of a single sample takes less than a second.

B. Recognition based on Temporal Data

HMMs are trained to estimate the likelihood L that the observed data o_i could have been generated by the model λ, i.e. $L = P(o_i|\lambda)$. In order to compare the predictions of two models, a threshold could be set for the difference between log-likelihoods, $log(L)$, of the two models. Therefore, for a sequence to be recognized as stable or unstable, the other models' predictions must be lower with respect to the chosen threshold in comparison. In Fig. 5(b,c), the performances are given with and without cross validation. Tab. III presents the chosen parameters for the corresponding experiments in Fig. 5(b,c) with 3 thresholds (the lowest, th_1, the middle, th_2, and the largest, th_3). Parameters are given in the format *(state (stable, unstable); mixture components (stable, unstable))*. It is evident that the classification rates are reasonable both with and without cross validation.

TABLE III

HMM CLASSIFICATION RATES AND PARAMETERS WITH th_1, th_2, th_3.

Node	Cross Validation		No Cross Validation	
	Rate	Parameters	Rate	Parameters
th_1	87.16%	(6, 3; 3, 5)	87.93%	(5, 5; 3, 5)
th_2	77.42%	(5, 2; 3, 4)	79.31%	(3, 2; 5, 5)
th_3	66.78%	(5, 2; 3, 4)	72.41%	(4, 2; 5, 5)

Fig. 6(a,b,c) are given with true positive rates against false positive rates to demonstrate how the HMM model parameters are chosen after training with different parameters. Each point in the figures indicates the performance of a trained

HMM pair while the red cross indicates the performance of the selected HMM pair. Different HMM models were trained with different numbers of mixture components and states. Finally, the best HMM pair was chosen based on the maximum classification rates for stable and unstable grasps. The blue lines cross where the classification performance gives equal number of false positives/negatives. The chosen HMM models result in a performance around this point as the best possible one among the trained models. The system can be adjusted so that the false positive rate is kept as low as possible while rejecting some stable grasps, since for our problem, classifying unstable grasps as stable grasps should be avoided as much as possible. We note that these results are based on the maximum recognition rates given the test set in order to see how good the classification could be.

To depict the difference on performance, the distributions of logarithms of likelihood ratios are presented for the three thresholds. Let L_s and L_u be the log likelihoods of the stable and unstable HMM models, then $r = L_s - L_u$ shows the log of the likelihood ratio. Fig. 7(a,b,c) show the histograms of these ratios (r) for stable and unstable samples. Green bars show the difference for stable samples and red bars are for unstable samples. We note that the green and red bars are more separate when the classification rate is higher. When the stability is more difficult to recognize, namely the classification rate is lower, there is more overlap.

VI. CONCLUSION

We have presented a system that integrates grasp planning and online grasp stability assessment based on tactile data on different types of objects. An important contribution of the presented work is an implementation and evaluation of the approach on a real robot system equipped with a three-fingered robot hand. We have analyzed the relation between analytical stability in simulation and real grasps under consideration of a range of uncertainties. In addition, we have applied an extended evaluation strategy based on different stability levels for the probabilistic learning technique. An extensive experimental evaluation demonstrates the feasibility of the approach and provides the bases for future implementation of the system for tactile exploration and closed loop control of corrective movements.

ACKNOWLEDGMENTS

This work was supported by EU through CogX, IST-FP7-IP-215181, and Swedish Foundation for Strategic Research.

REFERENCES

[1] C. Ferrari and J. Canny, "Planning Optimal Grasps," in *IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295.

[2] A. Miller and P. Allen, "Examples of 3D Grasp Quality Computation," in *Int. Conf. on Robotics and Automation*, 1999, pp. 1240–1246.

[3] C. Borst, M. Fischer, and G. Hirzinger, "Grasp Planning: How to Choose a Suitable Task Wrench Space," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 319–325.

[4] Y. Zheng and W. Qian, "Improving Grasp Quality Evaluation," *Robotics and Autonomous Systems*, vol. 57(6-7), pp. 665–673, 2009.

[5] A. Miller and P. Allen, "Graspit! A Versatile Simulator for Robotic Grasping," *Robotics and Automation*, vol. 11(4), pp. 110–122, 2004.

[6] J. A. Jørgensen, L.-P. Ellekilde, and H. G. Petersen, "RobWorkSim - an Open Simulator for Sensor based Grasping," in *Joint Conference of ISR 2010 and ROBOTIK 2010*, Munich, Germany, 2010.

[7] "OpenGRASP – A Simulation Toolkit for Grasping and Dexterous Manipulation," **URL:** http://opengrasp.sourceforge.net.

[8] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer, 2008, ISBN 978-3-540-23957-4.

[9] J. Tegin, S. Ekvall, K. Danica, J. Wikander, and I. Boyko, "Demonstration-based learning and control for automatic grasping," *Intelligent Service Robotics*, vol. 2, no. 1, pp. 23–30, 2009.

[10] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 1628–1633.

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

[12] 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.

[13] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An Experiment in the Use of Manipulation Primitives and Tactile Perception for Reactive Grasping," in *Robotics: Science and Systems, Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, 2007.

[14] M. Prats, P. Sanz, and A. del Pobil, "Vision-Tactile-Force Integration and Robot Physical Interaction," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3975—3980.

[15] L. Petersson, D. Austin, and D. Kragic, "High-level Control of a Mobile Manipulator for Door Opening," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 2333–2338.

[16] D. Kragic, L. Petersson, and H. I. Christensen, "Visually guided manipulation tasks," *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 193–203, 2001.

[17] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Bayesian Estimation for Autonomous Object Manipulation based on Tactile Sensors," in *IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 707–714.

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

[19] A. Jiméneza, A. Soembagijob, D. Reynaertsb, H. V. Brusselb, R. Ceresa, and J. Ponsa., "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.

[20] S. Chitta, M. Piccoli, and J. Sturm, "Tactile Object Class and Internal State Recognition for Mobile Manipulation," in *International Conference on Robotics and Automation*, 2010.

[21] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object Identification with Tactile Sensors using Bag-of-Features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 243–248.

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

[23] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using Entropy for Dimension Reduction of Tactile Data," in *14th International Conference on Advanced Robotics*, 2009.

[24] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, and V. Kyrki, "Learning Grasp Stability based on Haptic Data," in *Robotics: Science and Systems, Workshop on Representations for Object Grasping and Manipulation in Single and Dual Arm Tasks*, 2010.

[25] Y. Bekiroglu, V. Kyrki, and D. Kragic, "Learning Grasp Stability Based on Tactile Data and HMMs," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2010.

[26] K. Huebner, "BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping," in *International Conference on Intelligent Robots and Systems, Workshop on Grasp Planning and Task Learning by Imitation*, 2010.

[27] K. Huebner, K. Welke, M. Przybylski, N. Vahrenkamp, T. Asfour, D. Kragic, and R. Dillmann, "Grasping Known Objects with Humanoid Robots: A Box-based Approach." in *International Conference on Advanced Robotics*, 2009.

[28] L. R. Rabiner, *Readings in Speech Recognition*. Morgan Kaufmann Inc., San Francisco, 1990, ch. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296.

Learning Tactile Characterizations Of Object- And Pose-specific Grasps

Yasemin Bekiroglu Renaud Detry Danica Kragic

Abstract— **Our aim is to predict the stability of a grasp from the perceptions available to a robot before attempting to lift up and transport an object. The percepts we consider consist of the tactile imprints and the object-gripper configuration read before and until the robot's manipulator is fully closed around an object. Our robot is equipped with multiple tactile sensing arrays and it is able to track the pose of an object during the application of a grasp. We present a kernel-logistic-regression model of pose- and touch-conditional grasp success probability which we train on grasp data collected by letting the robot experience the effect on tactile and visual signals of grasps suggested by a teacher, and letting the robot verify which grasps can be used to rigidly control the object. We consider models defined on several subspaces of our input data – e.g., using tactile perceptions or pose information only. Our experiment demonstrates that joint tactile and pose-based perceptions carry valuable grasp-related information, as models trained on both hand poses and tactile parameters perform better than the models trained exclusively on one perceptual input.**

I. Introduction

This paper studies the exploitation of tactile, visual, and proprioceptive data for assessing stability in both planning and executing grasps.

Grasp planning relies on (1) the extraction of information from the agent's environment (e.g., through vision), and on (2) the recovery of memories related to the current environmental configuration (e.g., previous attempts to grasp a particular object). Because of the uncertainty inherent to these two processes, designing grasp plans that are guaranteed to work in an open-loop system is difficult. Grasp execution can thus greatly benefit from a closed-loop controller which considers sensory feedback before and while issuing motor commands.

Humans make extensive use of input from several sensor modalities when executing grasps [2, 3]. Clearly, vision is one of the modalities which contribute substantially to grasp control and stability [4, 5, 6]. Touch is another one, as supported by numerous studies which show the influence of tactile feedback on different grasp sub-processes [2, 3, 7, 1, 8]. For example, Johansson and Westling [7] have shown that anesthetizing a subject's fingers – thereby impairing his sense of touch while leaving his motor capabilities intact – directly leads to a loss in the subject's proficiency in grasping and lifting up objects. These observations are reflected in the work of Johansson et al. [2, 7, 1] who presented an empirical formalization of the human grasping behavior as a closed-loop system involving visual and touch feedback and a memory-based controller, see Figure 1. A key part of this work emphasized humans' ability to predict the repercussion of manipulative actions onto sensory channels by means of a (learned) forward model, thereby allowing us to react to unexpected situation and maintain grasp stability.

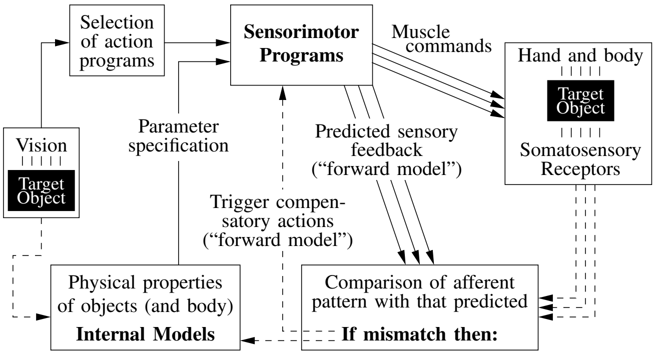

Selection of action programs
Sensorimotor Programs
Muscle commands
Hand and body
Target Object
Somatosensory Receptors
Vision
Target Object
Parameter specification
Predicted sensory feedback ("forward model")
Trigger compensatory actions ("forward model")
Physical properties of objects (and body)
Internal Models
Comparison of afferent pattern with that predicted
If mismatch then:

Fig. 1: Reproduction of "Predictive 'feed-forward' sensory control of manipulation", from R. Johansson [1], Fig. 5, p. 56. In Johansson's work [1], the sensorimotor planning and execution of manipulative tasks is formalized with a closed-loop system which integrates dynamic touch and visual perception with sensorimotor memories. A manipulative action is planned from visual input, which triggers an appropriate learned action program. As motor commands are being issued, sensor signals are continuously compared to values predicted by an internal forward model, which permits the detection of unexpected events. In turn, unexpected events trigger recovery procedures and the refinement of the forward model.

In robotics, vision-driven grasping and manipulation have been extensively studied [9, 10]. Vision has typically been used to plan grasping actions, and to update action parameters as objects move. Touch-based grasp controllers have also been studied, with emphasis on designing programs for controlling finger forces to avoid slippage and to prevent crushing objects [11, 12, 13].

In this paper, we discuss means of *learning* some of the models and sensorimotor programs which contribute to the system depicted in Figure 1. By observing the sensor signals issued during the execution of grasps demonstrated by a human, our agent learns what it feels like to grasp an object from a specific side, and learns which grasping configurations

Y. Bekiroglu, R. Detry and D. Kragic are with the Centre for Autonomous Systems, CSC, KTH, Stockholm, Sweden. yaseminb,detryr,danik@csc.kth.se.

This work was supported by the Swedish Foundation for Strategic Research, and by the EU project CogX (FP7-IP-027657).

lead to a stable grasp. When planning a grasp, the agent is able to compute an initial estimate of the stability of the planned grasp. As the grasp is being executed and the manipulator's fingers are brought around the object, the pose (3D position and 3D orientation) of the object is continuously tracked. When fingers come in contact with the object, afferent tactile signals are compared to the signals predicted by the learned feed-forward model for the current object-gripper configuration, yielding an updated estimate of the stability of the grasp. In a learning scenario, the agent can then proceed with an attempt to transport and shake the object to gather an empirical confirmation of its stability assessment, possibly updating the feed-forward model. During execution, if the stability estimate is too low, the agent may decide to move the manipulator to a better configuration before lifting up the object.

In mathematical terms, our agent learns an empirical representation of pose- and/or touch-conditional grasp success probability. This model predicts the stability of a grasp from tactile data and/or object-gripper pose parameters. We consider models defined on several subspaces of our input data – e.g., using only tactile perceptions or pose information. Models are optimized and evaluated with f-fold cross-validation. This model is presented in Section III.

To our knowledge, learning grasp controllers jointly from live visual and tactile feedback has not been attempted before. This experiment poses a number of technical challenges. As an object will often move while the robot is closing its hand to grasp it, the agent needs to track the pose of the object during the grasp, which is made difficult by the partial object occlusions effected by the robot hand. Section IV presents an overview of the robotic system we implemented to run our experiment.

We present an experiment that demonstrates that tactile perceptions carry valuable grasp-related information, as models trained on both hand poses and tactile parameters can perform better than the models trained exclusively on hand poses or tactile signals.

II. Related Work

Our work is related to vision-based grasp planning, tactile sensing, and robot learning.

Grasp planning is often approached by approximating object shape with a number of shape primitives such as boxes, cylinders, cones, spheres [14] and superquadrics [15] in order to limit the number of possible grasps and prune the search space to find stable grasps. Borst et al. [16] reduced the number of candidate grasps by generating random grasps dependent on the object surface and filtering them with a simple heuristic. Ciocarlie et al. [17] reduced the configuration space of a robotic hand to find pre-grasp postures from which the system searched for stable grasps. Grasp planning was also perfomed by using databases. Goldfeder et al. [18] demonstrated a grasp planner that used global similarity between a 3D model and the objects in a large database of 3D objects and grasps based on the intuition that similar objects were likely to have similar grasps. Li et al. [19] utilized a user-created database of human grasps. After a shape matching algorithm found the hand shape that best matched the query object, the alignment of the hand pose to the object shape was determined. The resulting candidate grasps were clustered and pruned depending on the task.

Learning aspects were considered in the context of grasping some of which focus on understanding human grasping strategies. Ekvall et al. [20] demonstrated how a robot system could learn grasping by human demonstration using a grasp experience database. The human grasp was recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup-table. Learning was also used to infer good grasping configurations based on visual input. Saxena et al. [21] introduced a system that learned grasping points by using hand labeled training data in the form of image regions which indicated good grasping points. A probabilistic decision system was then applied to previously unseen objects to determine a good grasping point or a region. Detry et al. [22] used vision to create grasp affordance hypotheses for objects and refined the grasp affordance hypotheses through grasping. The result was a set of grasps that would produce good grasps on a specific object. Erkan et al. [23] presented a probabilistic approach to model the success probabilities of grasp configurations obtained from visual descriptors and combined active and semisupervised learning to tackle the scarcity of labeled grasps.

In our work, we propose a system that learns to differentiate between successful and unsuccessful grasping configurations based on online visual and tactile feedback. The visual feedback is obtained by using a real-time tracker during grasping. We demonstrate the feasibility of our system on multiple objects.

Tactile sensing was used for various purposes in prior studies. For example, tactile sensing was used to maximize contact surfaces to remove a book from a bookshelf [24]. Application of force, visual and tactile feedback to open a sliding door was proposed by Prats et al. [25]. Tactile information was also used to determine object pose [26], the surface type (edge, flat, cylindrical, sphere) of the tactile contact [27] and deformation properties of objects [28]. Object shape was also extracted based on tactile feedback. Bierbaum et al. [29] performed a tactile exploration strategy with an anthropomorphic five-finger hand guided along the surface of previously unknown objects and a 3D object representation was built based on acquired tactile point clouds. Recognition of manipulated objects with tactile sensors was studied by using multiple grasp or manipulation attempts to learn the object shape from haptic signals. Schneider et al. [30] presented a bag-of-words approach to identifying objects using touch sensors available on a two-finger gripper. The approach processed tactile images collected by grasping objects at different heights. Gorges et al. [31] followed a similar approach for a humanoid hand by using features extracted from the tactile images in conjunction with the hand joint configurations as input data for the object classifier. Schöpfer et al. [32] used entropy to study the

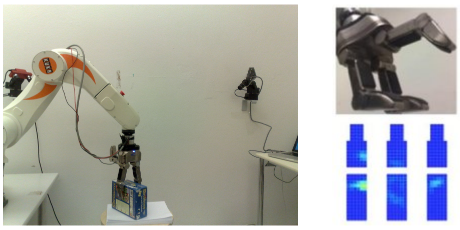


Fig. 2: Experimental robotic platform, composed of an industrial arm, a three-finger gripper equipped with tactile sensing arrays, and a camera. There are six tactile sensing arrays. The rightmost image shows an example of tactile readings obtained during a grasp.

performance of various features in order to determine the most useful features in recognizing objects using a plate covered with tactile sensor as the manipulator.

Differently from the aforementioned approaches, we are using tactile feedback to reason about grasp stability before further manipulating the object.

III. Learning Grasp Stability

Our aim is to infer grasp stability from the tactile imprints and the object-gripper configuration available *before* lifting up an object, and to provide the agent with means of learning from experience of how to make stability assessments.

A. *Perceptual Input*

Our robot platform is composed of an industrial arm and a three-finger hand, see Figure 2. Tactile imprints are delivered by pressure-sensing arrays deployed on the hand. Each of the hand's three fingers is composed of two segments, both covered by an array, yielding a total of 6 tactile arrays, see Figure 2. The tactile data is relatively high-dimensional and to some extent redundant. Therefore, we start by representing the acquired data as features. Here, we borrow some ideas from image processing and consider the two-dimensional tactile patches as images. We employ image moments as a suitable representation which also reduce the dimensionality. The general parameterization of image moments for one tactile array A is given by

$$m_{p,q} = \sum_i \sum_j ij A_{ij} \tag{1}$$

where p and q represent the order of the moment, and i and j represent the horizontal and vertical position on the tactile patch. We compute moments up to order two, $(p+q) \in \{0,1,2\}$, which yields 6 numbers that model the total pressure and the distribution of the pressure in the horizontal and vertical direction. We denote a tactile input vector by t. Such a vector contains moments from the six tactile pads and therefore holds 6×6 numbers.

Through visual and proprioceptive feedback, our platform is able to acquire object and gripper poses in real time. Gripper poses are simply obtained from the kinematics of the robot. Obtaining object poses is more challenging. As an object will often move while the robot is closing its hand to grasp it, the agent needs to compute the pose of the object *after* having closed the hand around it. This computation is made difficult by the partial object occlusions effected by the hand. Our aim however is not to get perfectly accurate pose information, but rather a rough idea of how the object is approached. We address this issue by tracking the movement of the object for the complete duration of the grasp. We are currently using a system which tracks the pose of a textured CAD model in a monocular video stream [33]. Tracking object textures greatly helps handling partial object occlusions and distractions induced by the hand.

We aim at designing a stability predictor that is independent of the position of an object. For this reason, we do not predict stability from the manipulator and object poses directly. Instead, we base our predictions on the *relative* object-manipulator pose. Object-relative manipulator configurations allow our system to encode notions such as "grasping a bottle from the side is better than grasping it from the top." However, stability will often not only depend on the relative object-gripper configuration, but also on the absolute orientation of the object. When an elongated object lies on a flat surface, it is generally better to grasp it close to its center of mass. Yet, if the object is standing, grasping it near its tip is acceptable. As a result, we also base our predictions on the angle between the gripper's approach vector and a direction aligned with gravity.

B. *Stability Classification*

We predict grasp stability with object-specific classifiers trained to discriminate between percepts that lead to stable or unstable grasps for a specific object. Our agent learns an empirical representation of pose- and touch-conditional grasp stability probability. This model is learned from a set of examples denoted by

$$Z = \{(x_i, y_i)\}_{i=1,\dots,n} \tag{2}$$

where each pair (x_i, y_i) is composed of perceptual readings $x_i \in \mathbb{R}^d$ (pose and touch) and a binary stability label $y_i \in \{\text{stable, unstable}\}$. Perceptual data are read during the execution of a grasping plan, shortly after the agent closed the manipulator's fingers around the object, but before any attempt to lift or transport the object. The probability of pose- and touch-conditional grasp stability is modeled with kernel logistic regression (KLR). In the next paragraph, we give an intuitive explanation of KLR applied to our problem. A short formal description follows. For further details on the theory behind logistic regression and kernel methods, we refer the reader to the work of Yamada et al. [34], Erkan et al. [23], and Schölkopf and Smola [35].

KLR models the stability probability of a grasp characterized by a perceptual vector x with the help of a weighted sum of the similarities between x and each vector in the training dataset Z. The weights associated to stable grasps will generally be positive, while those associated to unstable grasps will be negative. If x resembles percepts of Z that lead to stable grasps, its probability of stability will thus be high. In order to restrict values to the $[0,1]$ interval, KLR models probabilities by plugging the weighted sum described above into the logistic function $f(z) = \frac{1}{1+e^{-z}}$, which smoothly grows from 0 to 1 as its argument varies from minus infinity to infinity. Weights are usually chosen to maximize the probability of the training set.

Formally, we model the probability of pose- and touch-conditional grasp stability as

$$p(y = \text{stable}|x; v) = \frac{1}{1 + \exp\{-\sum_{i=1}^{n} v_i \mathcal{K}(x, x_i)\}} \tag{3}$$

where $p(y = \text{stable}|x)$ is the probability of success of a grasp characterized by the tactile and pose vector x, $\mathcal{K}$ is a kernel function that models the similarity between two perceptual readings and v is a weight vector chosen to maximize the regularized stability probability of the data

$$-\sum_{i=1}^{n} \log p(y_i|x_i; v) + c\,\text{trace}(vKv^T) \tag{4}$$

where K is the kernel Gram matrix, with $K_{ij} = \mathcal{K}(x_i, x_j)$, and c is a constant. This problem can be solved, e.g., with Newton's method. For more details, we refer the reader to the work of Yamada et al. [34]. In the experiments below, the constant c is chosen by cross-validation.

C. Kernel Function

As explained above, we consider perceptual signals in the form of tactile readings, relative object-gripper configurations, and an angle that represents the tilt of the hand's approach vector relative to gravity.

A vector x representing perceptual observations can be written as

$$x = (t, g, a) \tag{5}$$

where t is the tactile data, g is the object-relative gripper pose, and a is the angle between the approach vector and the vertical. The kernel $\mathcal{K}$ is defined as

$$\mathcal{K}(x_1, x_2) = \mathcal{K}_t(t_1, t_2)\mathcal{K}_g(g_1, g_2)\mathcal{K}_a(a_1, a_2). \tag{6}$$

The kernel function $\mathcal{K}_t$ simply corresponds to a multivariate isotropic Gaussian function

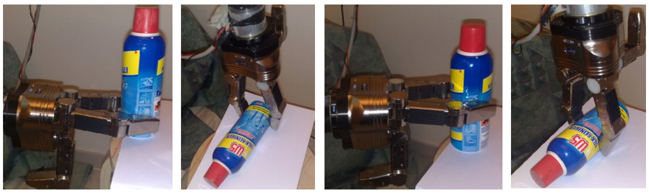
$$\mathcal{K}_t(t_1, t_2) = \mathcal{G}(t_1; t_2, \sigma_t), \tag{7}$$

where σ_t is a bandwidth parameter. In the next section, an optimal bandwidth is computed by cross-validation.

An object-relative gripper pose is composed of a 3D position and 3D orientation. We define the gripper pose kernel $\mathcal{K}_g$ as the product of a position and an orientation kernel. Let us denote the decomposition of a pose g into position and orientation by p and o respectively. We define $\mathcal{K}_g$ with

$$\mathcal{K}_g(g_1, g_2) = \mathcal{G}(p_1; p_2, \sigma_p)\frac{e^{\sigma_o\, o_1^T o_2} + e^{-\sigma_o\, o_1^T o_2}}{2} \tag{8}$$

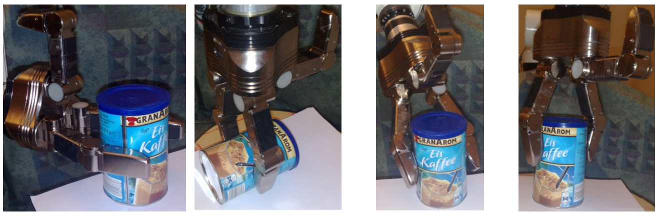
where $\mathcal{G}$ is a trivariate isotropic Gaussian kernel, the fraction corresponds to a pair of antipodal von-Mises Fisher distributions (Gaussian-like distribution on the rotation group [36, 37]), and the bandwidths σ_p and σ_o are fixed to allow for deviations of 20 mm and 20° respectively. For a more detailed mathematical description and motivation of $SE(3)$ kernels, we refer the reader to the work of Sudderth et al. [37].



(a) "Bottom" seed grasp

(b) "Middle" seed grasp

Fig. 3: Seed grasps for a detergent bottle. Each seed grasp is shown when the bottle is standing and lying.



(a) "Middle" grasp

(b) "Tilted"

(c) "Top"

Fig. 4: Seed grasps for a coffee pot.

The kernel function $\mathcal{K}_a$ corresponds to a Gaussian function

$$\mathcal{K}_a(a_1, a_2) = \mathcal{G}(a_1; a_2, \sigma_a), \tag{9}$$

where σ_a is a bandwidth parameter. In the next section, an optimal bandwidth is computed by cross-validation.

IV. Experiments

In this section, we present the perceptual data collected by the robot (392 grasps in total), and we discuss classification error rates for pose-based classification, tactile-based classification, and tactile-and-pose–based classification. We present in Section IV-A an experiment in which the agent explores grasping configuration around grasps demonstrated by a human. In Section IV-B, the agent tries grasps uniformly along one edge of an object.

A. Exploration around Demonstrated Grasps

We ran the first experiment on the two objects shown in Figure 3 and Figure 4. For each object, we demonstrated to the agent sets of two and three "seed" grasps that should be interesting to explore. Each of these grasps was parametrized by the pose of the hand with respect to the object. The agent was then tasked to explore the objects around these grasps. Each grasp trial worked as follows: An object was laid in front of the robot, at an arbitrary position. The standing/lying configuration of the objects also varied. For instance, Figure 3 shows the detergent bottle grasped when standing and when lying on the table. The agent estimated the pose of the object and selected one of the seed grasps available for that object.

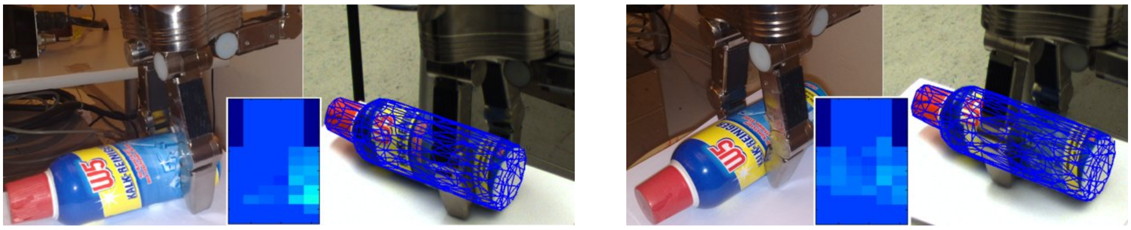


Fig. 5: Examples of grasps and associated tactile readings. Greyscale patterns correspond to the readings obtained from the frontmost distal array. The rightmost image of each image pair shows an overlay of the object's shape model aligned to the pose computed by the pose tracker [33].

Let us denote that grasp by g_s. In order to explore the object in the neighborhood of g_s, the agent generated a random grasp $\hat{g}_r$ from $P(g_r) \propto \mathcal{K}_g(g_s, g_r)$, where $\mathcal{K}_g$ is defined by Eq. 8. In effect, this lead the agent to explore grasps distributed a few millimeters/degrees away from g_s. The grasp g_r was executed by the robot. As the hand is rather big with respect to the objects, only two fingers were used. Grasping was run by simultaneously closing the fingers and applying a constant closing force on all joints. Once the hand had stopped, the agent recorded the pose of the object (which was usually different from the initial object pose) and the tactile imprints. It finally attempted to lift up the object. If lift-up could be achieved robustly, the grasp was marked as stable. If the object slipped or rotated in the hand while being lifted up, the grasp was marked unstable.

For the detergent bottle of Figure 3, seed grasps were defined at half height, and at the bottom end. Both seed grasps were explored in standing and lying configurations. The seed grasps of the coffee pot are shown in Figure 4. For standing configurations, all three seed grasps were explored. When lying on the table, only the "middle" was tried.

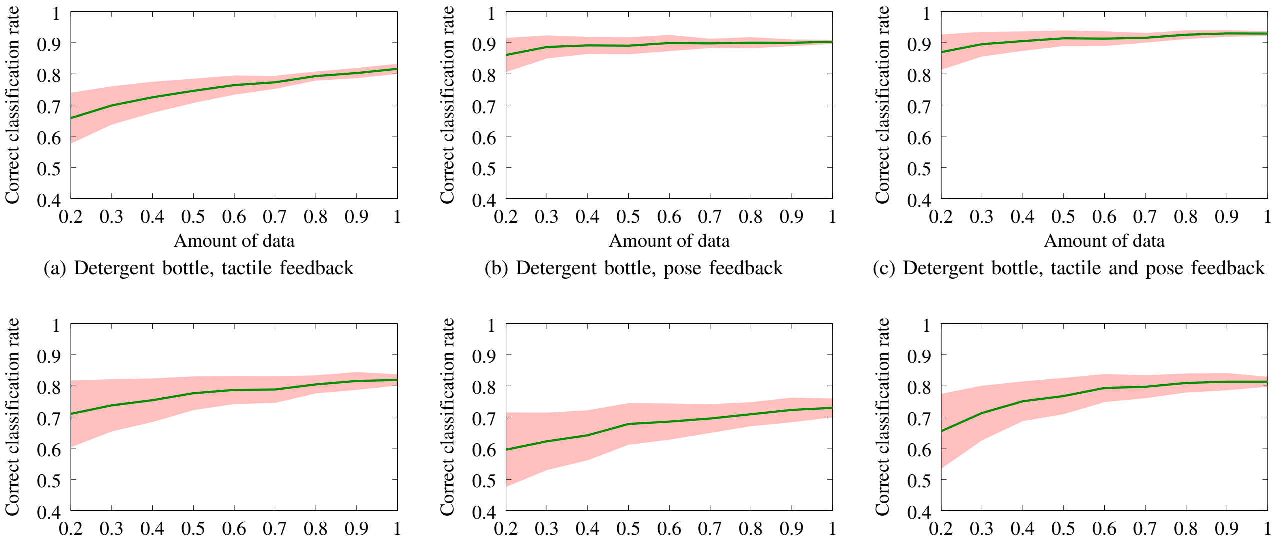
In this experiment, a total of 342 grasps were collected, i.e., 232 and 110 for the detergent bottle and the coffee pot respectively, with half of these stable and the other half unstable. Each grasp i consists of the grasp's perceptual readings t_i, g_i, a_i as defined by Eq. 5. For each object, tactile moments t_i were normalized to zero mean and unit variance. In order to evaluate the relevancy of tactile and visual feedback for stability estimation, we studied rates of correct classification for classifiers based on (1) tactile feedback alone, (2) pose feedback alone, and (3) both tactile and pose together. We note that as pose parameters cannot be shared across objects, each classifier is specific to one object – a classifier is learned and evaluated with the data collected for a single object. Stability classification was computed from the probabilistic stability model defined above (3). A grasp characterized by x was predicted to be stable if $P(y = \text{stable}|x) > \frac{1}{2}$. When classifying on tactile imprints or pose exclusively, the kernel of Eq. 6 was redefined as $\mathcal{K}(x_1, x_2) = \mathcal{K}_t(t_1, t_2)$ or $\mathcal{K}(x_1, x_2) = \mathcal{K}_g(g_1, g_2)\mathcal{K}_a(a_1, a_2)$ respectively. We computed success rates by ten-fold cross-validation. Cross-validation was run for several values of the tactile kernel bandwidth parameter σ_t (values between 0.5 and 5), and several values of the regularization constant c (see Eq. 4). Rates obtained with the best parameters are presented in Table I. For the detergent bottle, considering pose and tactile feedback jointly yields a higher classification rate than considering either pose or tactile alone. The bottle was explored around two seed points, both when standing and lying. When standing, both seed grasps lead to stable and unstable grasps. However, when lying, most grasps around the bottom of the bottle were unstable, while grasps around its center were both stable and unstable. Tactile feedback alone can difficultly make a difference between a grasp applied to the bottom of the bottle while it is standing or lying. For these grasps, the pose information (in the form of the angle of the grasp approach with the vertical) allows the classifier to separate stable and unstable grasps. For grasps applied around the center of the bottle, pose information allows the model to make reasonably good predictions, but taking tactile feedback into account refines these predictions.

	Detergent	Coffee pot
Tactile feedback only	82%	82%
Pose feedback only	90%	73%
Pose and tactile feedback	93%	82%

TABLE I: Correct classification rates from ten-fold cross-validation of three variants of the stability classification model for the detergent bottle and coffee pot.

For the grasps tried on the coffee pot, tactile feedback provides a better classification than pose, and considering both tactile and pose yields the same rate as tactile alone. The coffee pot is a rather light object compared to the detergent bottle. As a result of its low weight, the dependency of grasp stability on the standing/lying configuration of the object was less important than for the bottle. Tactile imprints however provided equally good stability assessments.

We also evaluated classification rates as a function of the amount of data available to the agent. Using fixed values for σ_t and c, we ran ten-fold cross-validations on increasingly large subsets of the collected data. We considered fractions of the data going from 20% to 100% of the total collected data, for each of which we ran multiple cross-validations. The mean classification rates are shown in green in Figure 6. These graphics show that for the detergent bottle, even small numbers



(a) Detergent bottle, tactile feedback

(b) Detergent bottle, pose feedback

(c) Detergent bottle, tactile and pose feedback

(d) Coffee pot, tactile feedback

(e) Coffee pot, pose feedback

(f) Coffee pot, tactile and pose feedback

Fig. 6: Rates of correct classification as a function of the available amount of data.

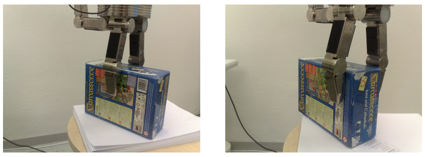


Fig. 7: Illustration of the grasps executed by the robot. The left image shows a "middle" grasp, which always succeeded, while the right image shows an "extremity" grasp which always failed. The object was continuously explored between these two points.

of examples allow for robust pose-based classification. For the coffee pot, collecting more data allows for more robust pose predictions. Red-shaded areas show one standard deviation. In all cases, the variance of the predictor decreases as the number of examples increases.

B. Exploration along the Top Edge of a Box

In the second experiment, the agent explored grasp poses distributed alongside the top side of a box (see Figure 7). Our aim with this experiment was to study how the transition between stable and unstable grasps occurred, and how accurately it was reflected by pre-grasp perceptual data. The wrist poses of the grasps executed by the robot were demonstrated by a human by teleoperation. The box was grasped by simultaneously closing the two fingers and continuously applying a constant closing force on all joints. A total of 50 grasps were executed, amongst which 25 were stable and 25 were unstable. Grasps applied near the middle of the top face of the box were always stable (see left image in Figure 7). As grasps were tried closer to the extremity of the box, they remained stable for a few centimeters, then abruptly became unstable. Unstable grasps were characterized by a rotation of the object when the robot tried to lift it up.

Stability classification was evaluated as explained above. Classification rates computed from tactile data alone yielded a 94% rate. Rates computed from pose data alone, and from pose plus tactile data, lead to 100% correct predictions. Several comments can be made on these results. First, pose perfectly separated stable grasps from unstable ones. We note however that in our setup, the camera is pointed directly at the objects, and the objects cover a large fraction of the camera's field of view. If the camera were to cover a larger field, such as the whole robot workspace, pose estimation would be less accurate, and pose-based classification would be less reliable, therefore motivating the use of additional perceptual modalities. Second, in this experiment, tactile imprints can discriminate surprisingly well grasps applied on both ends of the box. Although this discrimination may be useful in certain situations, it is likely that it is limited to the specific part of the box that was explored by the robot. As discussed below, one of our future aims is to learn models that characterize a *part* of an object, and which would thus be applicable to novel objects that share the same part. In this context, it will be interesting to study how tactile characterizations such as those learned for the box, or the objects of the previous section, generalize to novel objects.

V. Conclusion

This paper studied the viability of concurrent object pose tracking and tactile sensing for assessing grasp stability on a physical robotic platform. We presented a kernel-logistic-regression model of pose- and touch-conditional grasp success probability, and a robotic platform that can track the pose of an object while it is grasping it, and that can acquire tactile imprints of the grasps it executes. We showed that the robot is able to use data collected by human demonstrations to learn grasp stability classifiers. Our results showed that stability assessments based on both tactile and pose data can provide better rates than assessments based on tactile data alone.

Because models rely on the pose of an object, each model that the agent learns is only usable with that particular object. It is not realistic to imagine that an agent would learn different model of every object that exists. To overcome this limitation, we project to learn models that characterize only a *part* of an object, and which would thus be applicable to novel objects that share the same part.

VI. Acknowledgements

We warmly thank T. Mörwald, M. Zillich, and M. Vincze for their support with the Blort pose tracker [33]. We thank M. Yamada for providing us with his KLR library.

References

[1] R. Johansson, "Sensory input and control of grip," in *Novartis Foundation Symposium*, 1998, pp. 45–58.

[2] ——, "Sensory control of dexterous manipulation in humans," *Hand and brain: The neurophysiology and psychology of hand movements*, pp. 381–414, 1996.

[3] P. Jenmalm, S. Dahlstedt, and R. Johansson, "Visual and tactile information about object-curvature control fingertip forces and grasp kinematics in human dexterous manipulation," *Journal of Neurophysiology*, vol. 84, no. 6, p. 2984, 2000.

[4] R. Woodworth, "The accuracy of voluntary movement," *The Journal of Nervous and Mental Disease*, vol. 26, no. 12, p. 743, 1899.

[5] A. Milner and M. Goodale, *The visual brain in action*. Oxford University Press, USA, 2006.

[6] C. Hesse and V. Franz, "Grasping remembered objects: Exponential decay of the visual memory," *Vision Research*, 2010.

[7] R. Johansson and G. Westling, "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects," *Experimental Brain Research*, vol. 56, no. 3, pp. 550–564, 1984.

[8] A. Kritikos and C. Brasch, "Visual and tactile integration in action comprehension and execution," *Brain Research*, vol. 1242, pp. 73–86, 2008.

[9] B. Yoshimi and P. Allen, "Closed-loop visual grasping and manipulation," in *IEEE International Conference on Robotics and Automation*, 1996.

[10] D. Kragic, A. T. Miller, and P. K. Allen, "Real-time tracking meets online grasp planning," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 2460–2465.

[11] A. Bicchi, J. Salisbury, and P. Dario, "Augmentation of grasp robustness using intrinsic tactile sensing," in *IEEE International Conference on Robotics and Automation*, 1989.

[12] R. Howe, N. Popp, P. Akella, I. Kao, and M. Cutkosky, "Grasping, manipulation, and control with tactile sensing," in *IEEE International Conference on Robotics and Automation*, 1990.

[13] R. Howe, "Tactile sensing and control of robotic manipulation," *Advanced Robotics*, vol. 8, no. 3, pp. 245–261, 1993.

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

[15] 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.

[16] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2003, pp. 3692–3697.

[17] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," 2007.

[18] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *IEEE Intl. Conf. on Robotics and Automation*, 2009.

[19] Y. Li, J. L. Fu, and N. S. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 732–747, 2007.

[20] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 4715–4720.

[21] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[22] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Krueger, O. Kroemer, J. Peters, and J. Piater, "Learning continuous grasp affordances by sensorimotor exploration," in *From Motor Learning To Interaction Learning in Robots*, 1st ed., O. Sigaud and J. Peters, Eds. Berlin, Germany: Springer-Verlag, 2010.

[23] A. Erkan, O. Kroemer, R. Detry, Y. Altun, J. Piater, and J. Peters, "Learning probabilistic discriminative models of grasp affordances under limited supervision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1586–1591.

[24] A. Morales, M. Prats, P. Sanz, and A. P. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," in *Robotics: Science and Systems, Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, USA, 2007.

[25] M. Prats, P. Sanz, and A. del Pobil, "Vision-tactile-force integration and robot physical interaction," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3975—3980.

[26] 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.

[27] A. Jiméneza, A. Soembagijob, D. Reynaertsb, H. V. Brusselb, R. Ceresa, and J. Ponsa., "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.

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

[29] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann, "A potential field approach to dexterous tactile exploration," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2008.

[30] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *IROS'09: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 243–248.

[31] 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.

[32] M. Schöpfer, M. Pardowitz, and H. J. Ritter, "Using entropy for dimension reduction of tactile data," in *14th International Conference on Advanced Robotics*, IEEE. Munich, Germany: IEEE, Jun 2009.

[33] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT–the blocks world robotic vision toolbox," *Best Practice in 3D Perception and Modeling for Mobile Manipulation (Workshop at ICRA 2010)*, 2010.

[34] M. Yamada, M. Sugiyama, and T. Matsui, "Semi-supervised speaker identification under covariate shift," *Signal Processing*, vol. 90, no. 8, pp. 2353–2361, 2010.

[35] B. Schölkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.

[36] R. A. Fisher, "Dispersion on a sphere," in *Proc. Roy. Soc. London Ser. A.*, 1953.

[37] E. B. Sudderth, "Graphical models for visual object recognition and tracking," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.