ICT Call 7
ROBOHOW.COG
FP7-ICT-288533

Deliverable D4.2:

Multisensory exploration

April 15, 2014

Project acronym:	ROBOHOW.COG
Project full title:	Web-enabled and Experience-based Cognitive Robots that Learn Complex Everyday Manipulation Tasks
Work Package:	WP 4
Document number:	D4.2
Document title:	Multisensory exploration
Version:	1.0
Delivery date:	January 31st, 2013
Nature:	Report
Dissemination level:	Public
Authors:	Christian Smith (KTH) Yiannis Karayiannidis (KTH) Yasemin Bekiroğlu (KTH) Danica Kragic (KTH)

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n^{O}288533 ROBOHOW.COG.

Contents

Summary

The contribution in this deliverable is on multisensory object exploration and perception, as detailed in Tasks T4.3, T4.4, and T4.5. The ability to manipulate objects depends on knowledge of the objects' geometry, pose, and surface properties. Understanding both object properties and how these affect the interaction with the manipulator is necessary for using the objects as tools for performing subsequent tasks.
We explore different perceptual methods exploiting different sensory modalities, to gain understanding of what properties an object possesses and what types of manipulation action is possible for a specific object with a specific grasp. Algorithms have been developed both for detecting object properties and learning manipulation affordances.

Chapter 1
Published Results

The results for this deliverable have been accepted for publication in peer-reviewed venues. This section contains a short description of the contributions, and references to the published reports, which are appended to this document.

1.1 Multimodal Shape and Pose Perception

As detailed in task T4.4, we study perception of the geometric shape and pose of different objects, using a gaussian process framework to model geometric shape based on observations. The developed approach is initialized with a visual image of the object, and a series of tactile interactions is then performed to refine the model. The method includes a rational exploration strategy to determine the targets for tactile exploration that will reduce uncertainty the most. As detailed in task T4.3, we also study how we can combine external visual measurements of articulated objects (a robot) with a rough estimate of internal state (joint configuration) to refine a highly accurate model of that state using virtual visual servoing. The results are published in [1, 2].

1.2 Learning Manipulation Affordances

In the context of tasks T4.5, T2.3, and T5.4, we study the problem of measuring and learning manipulation affordances of different object/grasp combinations. We present a probabilistic framework for grasp modeling and stability assessment. The framework facilitates assessment of grasp success in a goal-oriented way, taking into account both geometric constraints for task affordances and stability requirements specific for a task. We also address the problem of identifying continuous bounds on the forces and torques that can be applied on a grasped object before slippage occurs. This is formulated as a regression problem which is solved using a Gaussian Process approach. We demonstrate a dual armed humanoid robot that can autonomously learn force and torque bounds and use these to execute actions on objects such as sliding and pushing. The results are published in [3, 4].

Bibliography

[1] M. Björkman, Y. Bekiroğlu, V. Högman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3180–3186.

[2] X. Gratal, C. Smith, M. Björkman, and D. Kragic, "Integrating 3d features and virtual visual servoing for hand-eye and humanoid robot pose estimation," in *IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 240–245.

[3] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 3040–3047.

[4] F. Vina, Y. Bekiroglu, C. Smith, Y. Karayiannidis, and D. Kragic, "Predicting slippage and learning manipulation affordances through gaussian process regression." in *IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 462–468.

Enhancing Visual Perception of Shape through Tactile Glances

Mårten Björkman, Yasemin Bekiroglu, Virgile Högman, and Danica Kragic

***Abstract*— Object shape information is an important parameter in robot grasping tasks. However, it may be difficult to obtain accurate models of novel objects due to incomplete and noisy sensory measurements. In addition, object shape may change due to frequent interaction with the object (cereal boxes, etc). In this paper, we present a probabilistic approach for learning object models based on visual and tactile perception through physical interaction with an object. Our robot explores unknown objects by touching them strategically at parts that are uncertain in terms of shape. The robot starts by using only visual features to form an initial hypothesis about the object shape, then gradually adds tactile measurements to refine the object model. Our experiments involve ten objects of varying shapes and sizes in a real setup. The results show that our method is capable of choosing a small number of touches to construct object models similar to real object shapes and to determine similarities among acquired models.**

I. Introduction

One of the reasons that makes the process of autonomous grasping challenging is that object properties required for grasp planning such as shape are commonly not known a priori. In addition, sensory information used to extract this information from the environment, e.g. vision, is prone to error. Processes prior to shape extraction such as scene segmentation are not perfectly accurate due to several issues, e.g., occlusions and noisy measurements. Besides object shape, conceptual high-level object category information is another important input that can be used. In particular, for goal-oriented grasp planning, different instances from the same category can be grasped in a similar way for a particular task. For instance, bottles should be grasped from a side for a pouring task, so as not to block the opening.

Humans interact with the environment using rich sensory information. Studies show that both visual and haptic modalities contribute to the combined percept [1]–[3]. Results from [3] suggest that observers integrate visual and haptic shape information of real 3D objects and that bimodal shape estimates are more reliable than shape estimates that rely on either vision or touch alone.

The goal of our work is to complement visual information with tactile sensing in order to acquire 3D object models. We investigate how to deal with uncertainties in the sensory data to extract object shape and category. Given a scene like the one shown in Fig. 1, with an object in the center of view, our goal is to gain insight on what manipulation actions the object affords. If the shape of the object were known, one could get some idea of what actions to consider, especially if the shape is similar to an object that has already been manipulated. Much information can be provided through stereo vision, using e.g. a Kinect device. Regardless of which stereo vision system is used, however, only one side of an object is seen, i.e., the one side facing the cameras. Without any additional sensory modalities, one can only make qualified guesses of what the occluded side looks like, using assumptions such as symmetry [4], assumptions that may well be incorrect. In this paper we instead propose touch as a means to get additional information. By carefully touching the object, we will show how an object model can be created, a model that provides enough information to categorize the object based on shape.

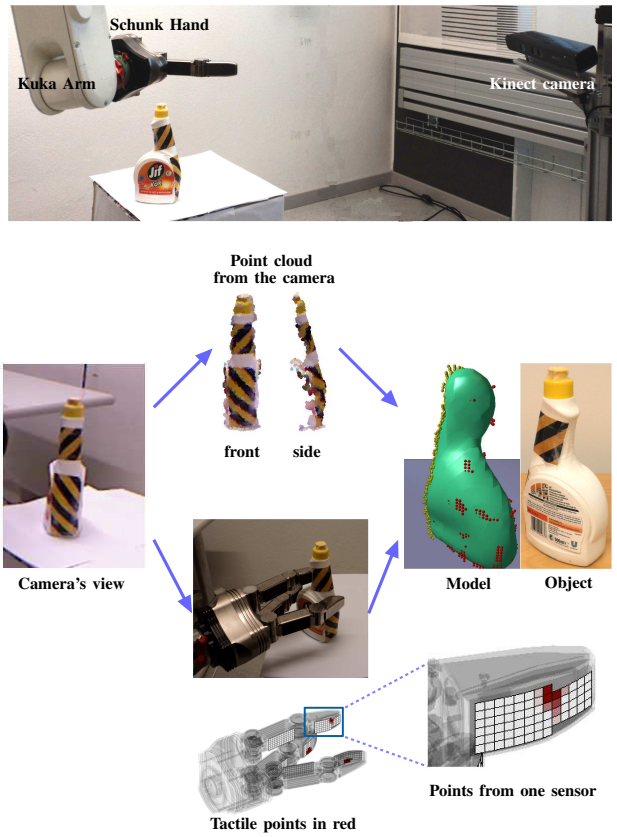


Fig. 1: Extracting object model: We rely on visual measurements from a Kinect and tactile measurements from the fingers. The model is formed based on the point cloud (in yellow) from the camera and the contact points (in red).

M. Björkman, Y. Bekiroğlu, V. Högman, D. Kragic are with the Centre for Autonomous Systems and the Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {celle|yaseminb|virgile|dani}@kth.se. This work was supported by the Swedish Research Council and the EU projects eSMCs (FP7-IST-270212) and RoboHow.Cog (FP7-ICT-288533).

II. Related work and Contributions

In robotics, object shape estimation has been studied with unimodal data, i.e., only visual [5] or tactile [6] sensing, and bimodal data with visual and tactile sensing combined [7]. Clearly, vision alone delivers useful information about object shape. Krainin et al. [5] proposed a method where a robot picks up and moves an object in front of a sensor. Their approach based on Kalman filters is able to build 3D models of unknown objects using a depth camera observing the robot's hand moving the object. However, they showed that the approach may produce failures with poor alignment in case of a combination of high uncertainty in the object pose, nondistinctive object geometry (completely planar surface) or fairly uniform color and poor lighting conditions. Tactile information can be used to alleviate such problems.

Bierbaum et al. [8] introduced the idea of using Dynamic Potential Fields for tactile exploration to build a contact/tactile point cloud of an unknown object. Their system requires a rough initial estimate about the object position, orientation and dimension, then exhaustively performs grasps in unexplored regions. Faria et al. [9] also builds contact point clouds in an exhaustive way. They however follow a probabilistic approach to store the extracted tactile points in a volumetric map. In their experiments, a human subject wearing a glove with magnetic tracking sensors to obtain fingertip positions performs grasps that follow the contour of objects. Meier et al. [6] followed a similar strategy and used a probabilistic approach, Kalman filters, to build a model of the contact point cloud. Their robot grasps objects at different heights and positions also varying the orientation of the hand. Their results show that the acquired models can successfully be used for classification.

There are approaches that supplement vision with more sensory information especially where visual sensing is weak, e.g., occluded object parts. Maldonado et al. [10] used a proximity sensor to scan the unseen parts of an object by a depth camera without touching the object. They combined the point cloud from the camera and the sensor and built a 3D Gaussian point representation based on the convex hull of the complete point cloud. Their representation simply contains the centroid and the shape of the object through the mean and the covariance matrix of the Gaussian distribution. Dragiev et al. [7] has included laser data in addition to haptic measurements in order to complement vision. They proposed to use Gaussian Process Implicit Surfaces to fuse the uncertain sensory data and showed that this representation can be used to control reaching and grasping such that the hand is moved and oriented towards the object and grasps aligning the fingers according to the object shape. There are also studies that focus on object recognition without explicitly modeling the full 3D shape, but rather representing the objects based on visual [11], tactile [12], [13] or both features [14].

Differently from the aforementioned approaches, we focus on building object models that can be extracted with a small number of actions (touches) in order to understand the category objects belong to, rather than exhaustively trying to explore the whole object. This is an iterative process where the robot executes more touches as it is less confident about the object shape. In summary, our contributions can be listed as follows:

- We incrementally include tactile readings in the shape estimation to further refine the object model that is initialized based on visual measurements only.
- We use a probabilistic approach to shape estimation through Gaussian Process regression to deal with uncertainties in sensory measurements.
- Instead of an exhaustive exploration, we obtain a model of a given object by selecting where to touch next, given the object regions where the shape estimation is most uncertain.
- Our system is able to build models that can be used for shape categorization after a small number of touches.

III. Object modelling

In this section, we describe how objects are represented based on visual and tactile measurements. We introduce Gaussian Process regression modeling of Implicit Surfaces, the strategy to determine how to acquire tactile data and the shape descriptors used for measuring similarities between different objects.

A. Visual Measurements

An observed object is segmented from its background using a segmentation and tracking system that works over sequences of touches. The system uses stereo vision, in our case a Kinect device, in an heterogeneous MRF based framework [15]. The framework uses color and depth information to divide the scene into either planar surfaces, bounded objects or uniform clutter models. The planar and uniform models are automatically initialized, while an ellipsoid used to model the observed object is initiated by a point that is manually placed inside the corresponding image region. From the resulting object segments we get point clouds that serve as starting points for object modelling. Later we will complement these points with tactile readings from touches.

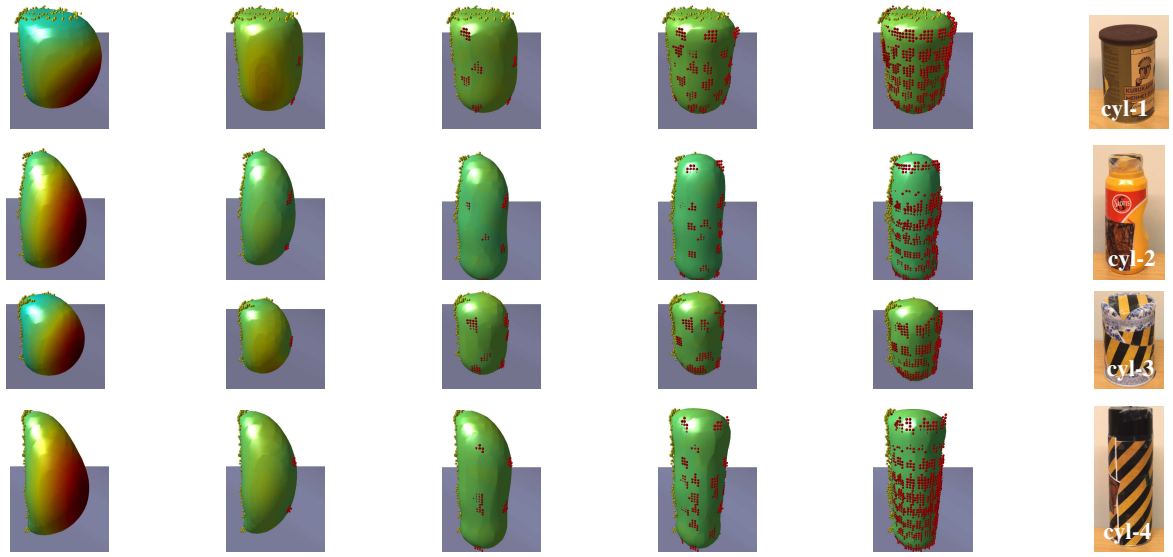
B. Implicit surfaces

From a set of measurements of 3D points $\{\mathbf{x}_i, i = 1...N\}$ that are located on the surface of an object, we now describe how to derive implicit surfaces for representation. The model should later be used for deciding object category based on shape. In our case the measurements originate from stereo vision as well as tactile readings. With a function $f : \mathbb{R}^3 \mapsto \mathbb{R}$, we define an implicit surface by the supporting points $\mathbf{x} \in \mathbb{R}^3$ that satisfy

$$f(\mathbf{x}) = 0.$$

The function $f(\mathbf{x})$ is modelled by Gaussian Process (GP) regression [16], with each observation $y = f(\mathbf{x}) + \epsilon$ assumed to be subjected to zero-mean Gaussian noise, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. The shape of the GP is governed by a thin plate covariance function [17]

$$cov(f(\mathbf{x}_i), f(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j) = 2|r|^3 - 3Rr^2 + R^3,$$



(a) v. only (b) 1 t. (c) 4 t. (d) 12 t. (e) 54 t. Real

Fig. 2: Evolution of object models against the number of touches (t.) for the cylinders: (a) Initial models based solely on visual (v.) measurements depicted by yellow points. The models are oriented to show the back sides that are not visible by the camera. Highest uncertainty is represented by red color and dark green regions correspond to least uncertainty. (b) Models after including tactile measurements from one touch applied to the region with highest uncertainty. Contact points obtained by touching are depicted by red points. (c) Models after 4 touches. (d) Models after 12 touches which were found sufficient to group all the objects confidently. (e) Models after exhaustively exploring the objects which require 54 touches with our setup. (f) The real objects for comparison.

where $r = |\mathbf{x}_i - \mathbf{x}_j|$ and R is a maximum possible value of r. This covariance function has slightly better characteristics than the more frequently used squared exponential function, in particular for rectangular objects where the flatness of surfaces needs to be preserved. Quantitatively, however, we have not observed any significant differences between the two when applied for categorization.

The model is learned from a set of tuples $(\mathbf{x}_i, y_i)$, where $y_i = 0$ for the stereo vision or tactile measurements. Since a physical object, at least those that can be acted on by a robot, occupies a certain volume in 3D space, the implicit surfaces need to be compact (closed and bounded). In order to guarantee this, we place additionally exterior points, for which $y_i = +1$, on the boundaries of the scene and a single interior point that is forced to be inside the closed surface with $y_i = -1$.

In the later experiments, this interior point was chosen as the centroid of the stereo point cloud displaced by 1 cm along the direction of the camera, assuming that this is the smallest object thickness one can expect. With objects assumed to be located within a cube with side lengths $L = 30$ cm and centered at the centroid, the parameter R was set to $\sqrt{3}L$. The only remaining hyperparameter is the expected noise level which is set to $\sigma_n^2 = 0.1$. The value was chosen so as a balance between the smoothness of the surfaces and the noise in the integration of tactile and visual readings.

C. Action selection and cue integration

With GP regression we do not explicitly get a function $f(\mathbf{x})$, but the mean $\bar{f}$ and variance $\mathbb{V}(f)$ of all possible functions that could fit the measurements. The variance can be used as a measure of uncertainty, with higher variance for points far away from already recorded measurements. Examples of implicit surfaces and variances can be seen in Fig. 2a. A surface is given by points for which the mean is zero and the colors illustrate the corresponding variances, with red for points of highest variance. The stereo vision point clouds are shown as yellow points, most of which are occluded by the objects in the figure.

To refine the object models and decrease the uncertainty, the robotic hand is guided towards those points for which the variance is large in order to select a position for touching. We call these touches *ordered* touches in the experiments below, as opposed to *random* touches where new touches are selected in random order. The arm-hand configuration has earlier been calibrated with respect to the camera system, with a precision of a few millimeters. The highest variance point is searched for in a discrete action space defined by the vertical position and the approach angle, both of which are computed with respect to the centroid of the current model. For each possible action, the closest point to each respective tactile sensor pad is found on the implicit surface. The action selected for execution is then based on the maximum variance found among all actions and sensor pads.

Touches are then executed in sequence and the GP model

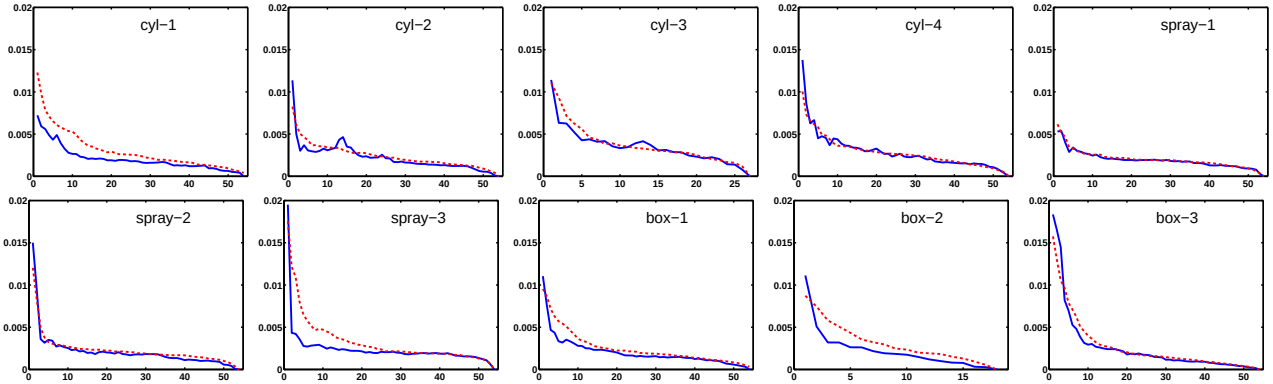


Fig. 3: Convergence of curvature point clusters using ordered (solid) and random (dashed) touches with respect to the final result after up to 54 touches (x-axis). Plots from random touches are based on the average from 10 runs and are thus smoother than those of the ordered touches.

is updated accordingly, as new measurements are integrated with the model. To speed up computations, the measurement set is made sparser to about two thousand points[1]. This is necessary since the computational cost of the GP increases to the cube of number of points. Recorded tactile measurements can be seen as red points in Fig. 2 for an increasing number of touches. In some cases, these points are displaced with respect to the implicit surface, which might happen if the object moves considerably when it is touched. To minimize these displacements, which is critical for long sequences of touches, the object frame is constantly updated for each new touch. This is done by registering the stereo vision point clouds, given by the segmentation system, before and after a touch using the Iterative Closest Point algorithm [18] and transforming new measurements back to the original frame.

D. Shape Descriptors

Representing object shape as a GP or a mesh derived from points on the resulting implicit surface is not straightforward, if the goal is to compare shapes for action selection. Instead we represent the extracted implicit surfaces with shape descriptors that capture information invariant to possible manipulation actions, while discarding redundant information. Two very different objects may afford similar actions, while two seemingly similar objects might not. For example, a rectangular box and a cylinder typically require different grasping strategies, but they may well appear similar when e.g., represented as ellipsoids, if aspect ratios are similar.

In this work, we look at two different rotation and translation invariant shape descriptors; 3D Zernike moments and surface curvatures. Zernike moments have successfully been used for shape retrieval [19] and are attractive due to the flexibility and low number of dimensions required, as well as the fact that Euclidean distances can be used for shape comparison. For the Zernike moments, voxelization is first applied in a 3D grid with voxels of side length $l = 0.75\ cm$, keeping the interior voxels for which the GP means are $\bar{f} \leq 0$ at their center points.

[1]On a 3.2 MHz Core i7 CPU the cost of computing the GP model and associated shape descriptor is about 4 s using PCL, VTK and Eigen.

For comparison using surface curvatures, the Marching Cubes algorithm [20] is first applied to the same grid to find a triangular mesh representing the implicit surface. From this mesh, principal curvatures are then computed [21], with one 2D measurement per vertex point. The shape of an object is thus represented by a sample set of about 500 measurements of curvatures. A kernel based two sample test [22] is used to compare two such representations, using Gaussian kernels with standard deviations of 0.25, which yields a soft decision on the similarity between the sample sets.

IV. Experimental Evaluation

In this section, we first describe our experimental platform and then present results from shape estimation and categorization experiments comparing touch selection strategies and shape descriptors.

The experimental robot platform is composed of an industrial Kuka arm (6 dof), a three-finger Schunk Dextrous hand (7 dof) equipped with tactile sensing arrays, and a Kinect stereo vision camera. The robot can acquire tactile imprints via pressure sensitive tactile pads mounted on the Schunk hand's fingers. Each finger of the hand has 2 tactile sensor arrays composed of 6x13 and 6x14 cells, which yields at most 486 tactile points after one touch. For each touch, the hand is set to a fixed initial joint configuration where the thumb opposes the other two fingers as seen in Fig. 1, then fingers are closed until contact is sensed.

In an earlier study [23] we concluded that the object class was an important factor, if one wants to determine what grasping action to pursue to fulfill particular tasks, tasks such as hand-over, pouring or dish-washing. However, the object class was not derived directly from sensory data, but given manually prior to the experiments. In this work, we aim to automate this process by learning shape-dependent features to replace the manually set object class. Our starting point is thus a set of objects for which we know the respective affordances from earlier experiments. These ten objects can be seen in Fig. 4, with names indicating the similarity in afforded actions. The end goal is to use stereo vision and tactile measurements through a series of touches to

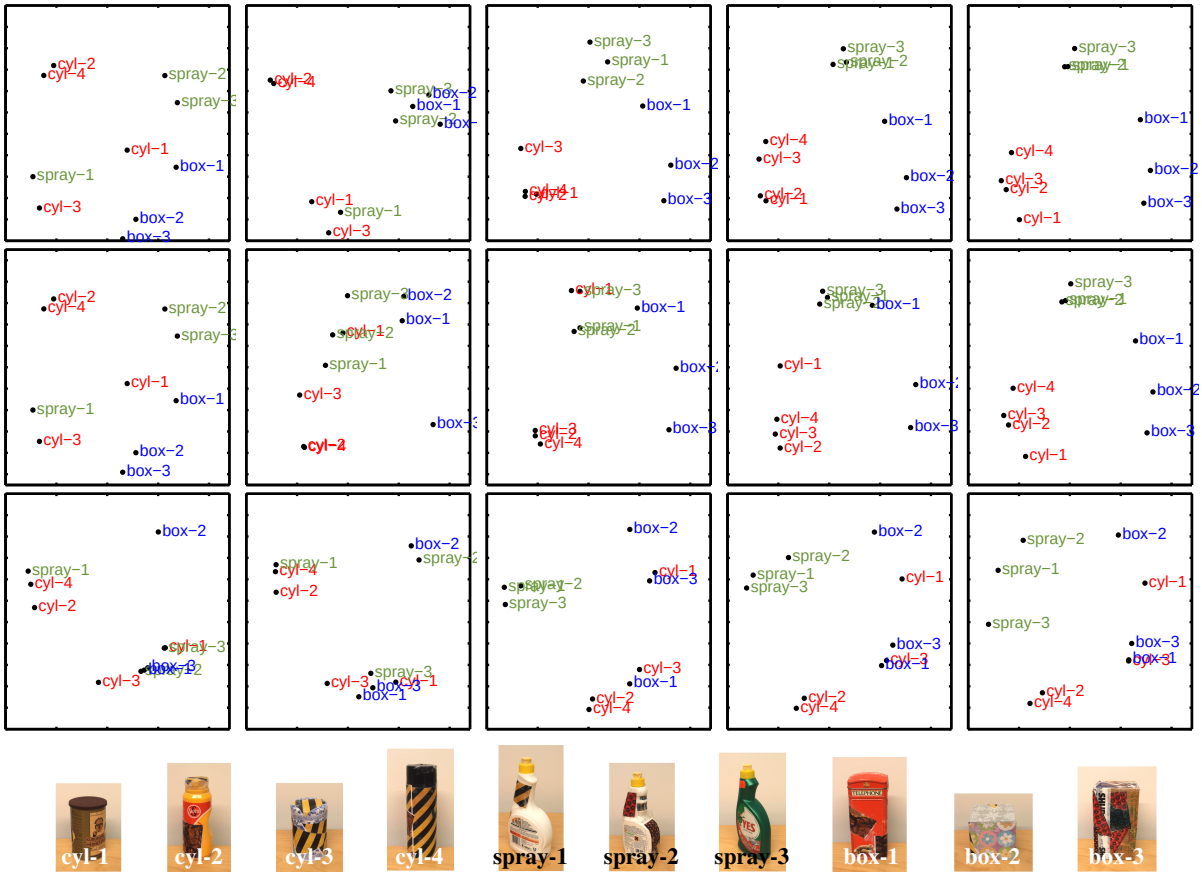


Fig. 4: Spectral embeddings of curvature point clusters, after 0 (left), 1, 4, 12 and 54 (right) touches, using ordered (first row) and random (middle row) touches, as well as with Zernike moments and ordered touches (last row). Ordered touches lead to faster convergence than random touches, and Zernike moments cluster objects more based on similarity in object aspect ratios, than similarities in affording grasp actions.

determine which grasping action the object would afford. The question is: how many touches this would require and what representation should one aim for?

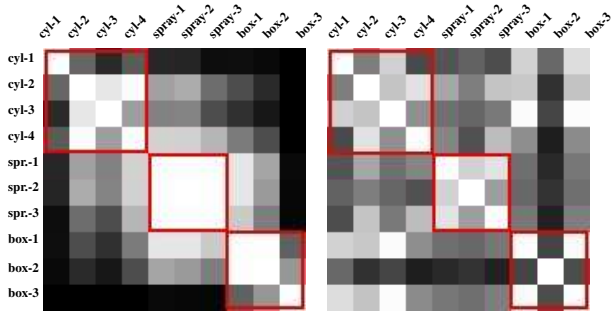
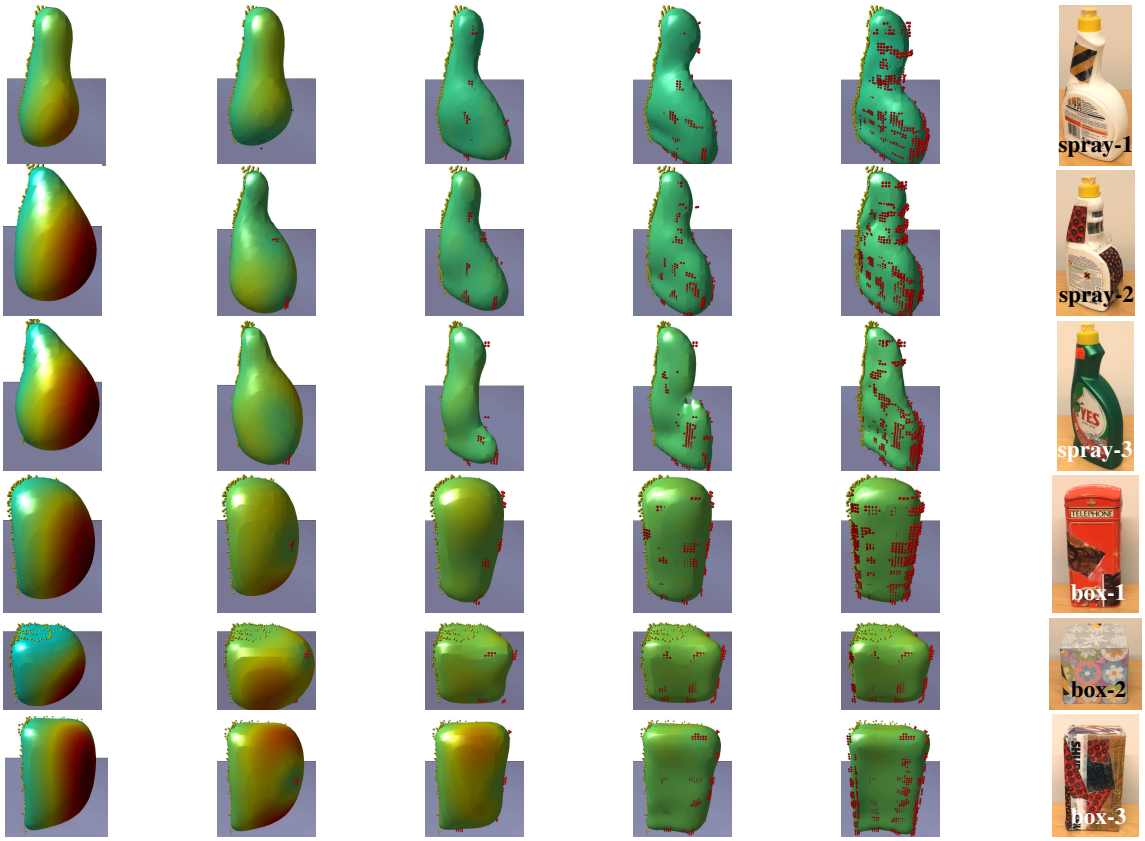


Fig. 5: Similarity matrices using up to 54 touches with columns and rows given by the objects in Fig. 4 using either curvature measures (left) or Zernike moments (right).

A. *Experimental results*

The ten objects were placed on a table-top with the Kinect camera overlooking objects from one side. To fully cover an object with tactile measurements, up to 54 touches (27 for *cyl-2* and 18 for *box-2* due to their lower heights) were performed from the side parallel to the table in a grid of 9 angles (22.5° apart) and 6 heights (spaced at a vertical distance of 2 cm) with respect to the table. The tactile measurements are illustrated as red points in Fig. 2. From the resulting implicit surface model, shape descriptors based on curvatures and Zernike moments (up to order 10) were computed and analyzed.

The convergence of the curvature based descriptors was studied by computing the distances between the descriptors after different numbers of touches and the final one. In Fig. 3 the convergence is shown using either ordered touches computed from points of maximum GP variance or touches selected randomly. The randomly generated sequences of touches were executed 10 times and then averaged. Thus the corresponding curves are slightly smoother than those of the ordered touches. The difference between the two strategies is not consistent. For most objects the difference is small and for some objects random touches are sometimes better, in particular in the beginning. The reason is because ordered pushes are computed from implicit surfaces obtained so far and at an early stage the shapes are still mostly unknown. For



(a) v. only (b) 1 t. (c) 4 t. (d) 12 t. (e) 54 t. (f) Real

Fig. 6: Evolution of object models against the number of touches for the spray bottles and the boxes. See Fig. 2 for details.

the box shaped objects, the first ordered push is usually at a corner edge on the back side of the object, when a preferable push would instead have been on one of the sides. Thus it takes another push or two for the ordered touches to catch up. From the graphs in Fig. 3, as well as from Fig. 2 and 6, it can be concluded that most changes occur during the initial ten touches.

As an illustration of the similarity between different objects, similarity matrices were computed for both curvature and Zernike based shape descriptors, which are given in Fig. 5. From the structures of the two matrices it can be concluded that while the curvatures capture classes relevant for grasping, Zernike moment does not do so to the same degree. In fact, the grouping is quite different for Zernike moments and more related to the aspect ratios of the objects than the curvatures.

This can be more easily illustrated with spectral clustering. Using the method of Ng et al. [24], we computed 2D spectral embeddings from the similarity matrices, embeddings that are shown in Fig. 4 for different numbers of touches. Here the object *cyl-3* is grouped with *box-1* and *box-3* for Zernike moments, due their similar height/width ratios. Whereas the elongated *cyl-2* and *cyl-4* are similar, they are very different from the shorter cylinder *cyl-1*. Even if *box-1* is a bit distant from *box-3* using curvature measures, the three classes can still be trivially found using e.g. k-means clustering. From the embeddings, the benefits of ordered touches can also be seen, compared to the random ones. Already after four touches, the three classes are grouped, even if it is not until 12 touches the *box-1* is closer to the other boxes than the group of spray bottles. The reason for this is that this box is thinner than the other boxes and since the GPs tend to smoothen edges, it is more like a spray bottle after too few touches. The thin plate prior tends to weaken this effect compared to a typical exponential one.

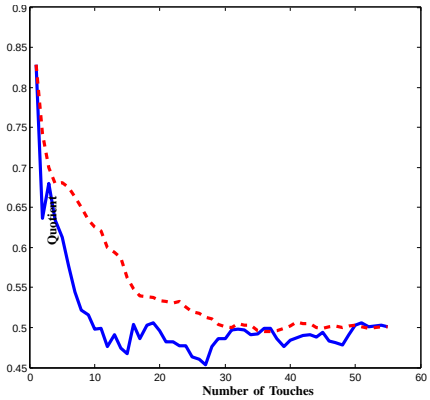


Fig. 7: Evolution of quotient between within- and between-category distances with random (dashed) and ordered (solid) touches using curvatures, as the number of touches increases.

A final illustration of the benefits of ordered touches for shape discrimination can be seen in Fig.7, where the quotient between within- and between-category distances are shown for an increasing number of touches. The quotient stabilizes after only about ten ordered touches, but for random touches at least 25 touches are required. Thus even if the benefits of ordered touches are sometimes limited when studying individual objects, they are considerable for categorization.

V. Conclusions

This paper has presented a method [2] for creation of object models from visual and tactile measurements, with the goal of later applying these for classification and manipulation. From an initial set of visual measurements, an object model is refined by touching the corresponding object on surface points predicted to be most uncertain. Given a curvature based representation of object shape, it was shown that about ten touches are sufficient for objects to be grouped into clusters relevant for manipulation. What remains to be tested in future work, however, is to what extent this representation captures manipulation affordances and can be directly used for action selection, preferably without using an intermediate step of supervised object classification.

A weakness of the current system arises from the fact that GPs have a computational cost proportional to the number of measurement points cubed. To cope with this we currently sample from the total set of points to make the problem computationally tractable. However, there are methods for sparse GPs that choose an optimal subset of points instead [25], [26], which will become a necessity in particular if measurements from additional modalities are later included.

The presented work can be extended in several directions. We intend to investigate more descriptors, other than surface curvatures and Zernike moments, that can be useful for object categorization. We will further integrate the presented approach with a pushing mechanism that can provide additional information on object affordances, e.g. rolling or sliding, potentially leading to more informed decisions about whether more measurements are needed given a particular task. Grasp planners e.g., often need information on object category [23], [27] to plan goal-directed grasps, where objects from the same category can be grasped in a similar way. Hence, we also plan to test the obtained object models for grasping tasks by using them for grasp planning.

References

[1] N. Gaissert and C. Wallraven, "Integrating visual and haptic shape information to form a multimodal perceptual space," in *IEEE World Haptics Conference (WHC)*, June 2011, pp. 451–456.

[2] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.

[3] H. B. Helbig and M. O. Ernst, "Optimal integration of shape information from vision and touch," *Experimental Brain Research*, vol. 179, no. 4, pp. 595–606, 2007.

[2] A video that illustrates the method can be found at: `http://www.csc.kth.se/~celle/`

[4] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 686–693.

[5] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3d object modeling," *Int. J. Robotics Research*, vol. 30, no. 11, pp. 1311–1327, September 2011.

[6] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 630–635, June 2011.

[7] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 2845–2850.

[8] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann, "A potential field approach to dexterous tactile exploration of unknown objects." in *IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 360–366.

[9] D. R. Faria, R. Martins, J. Lobo, and J. Dias, "Probabilistic representation of 3d object shape by in-hand exploration," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2010, pp. 1560–1565.

[10] A. Maldonado, H. Alvarez-Heredia, and M. Beetz, "Improving robot manipulation through fingertip perception," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Vilamoura, Algarve, Portugal, 2012, pp. 2947–2954.

[11] A. Ude, D. Omrcen, and G. Cheng, "Making object learning and recognition an active process," *Int. J. Humanoid Robotics*, vol. 5, no. 2, pp. 267–286, 2008.

[12] 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 Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 243–248.

[13] Z. A. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager, "Tactile-object recognition from appearance information," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 473–487, 2011.

[14] P. Allen, "Integrating vision and touch for object recognition tasks," Dept. of Computer Science, Columbia University, Tech. Rep., 1986.

[15] M. Björkman and D. Kragic, "Active 3D segmentation through fixation of previously unseen objects," in *British Machine Vision Conference*, August 2010, pp. 119.1–119.11.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[17] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," in *Gaussian Processes in Practice Workshop*, 2007.

[18] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[19] M. Novotni and R. Klein, "Shape retrieval using 3d zernike descriptors," *Computer-Aided Design*, vol. 36, no. 11, pp. 1047–1062, 2004.

[20] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM Siggraph Computer Graphics*, vol. 21, no. 4, 1987, pp. 163–169.

[21] X. Chen and F. Schmitt, "Intrinsic surface properties from surface triangulation," in *European Conference on Computer Vision (ECCV)*. Springer, 1992, pp. 739–743.

[22] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[23] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[24] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[25] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural Computation*, vol. 14, no. 3, pp. 641–668, 2002.

[26] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2009, pp. 567–574.

[27] M. Madry, D. Song, and D. Kragic., "From object categories to grasp transfer using probabilistic reasoning," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, Minnesota, USA, 2012, pp. 1716–1723.

Integrating 3D Features and Virtual Visual Servoing for Hand-Eye and Humanoid Robot Pose Estimation

Xavi Gratal, Christian Smith, Mårten Björkman and Danica Kragic

***Abstract*— In this paper, we propose an approach for vision-based pose estimation of a robot hand or full-body pose. The method is based on *virtual visual servoing* using a CAD model of the robot and it combines 2-D image features with depth features. The method can be applied to estimate either the pose of a robot hand or pose of the whole body given that its joint configuration is known. We present experimental results that show the performance of the approach as demonstrated on both a mobile humanoid robot and a stationary manipulator.**

I. Introduction

Most of the object grasping and manipulation tasks require the pose between the robot hand and the object to be known prior to or during execution of the grasp. Although power grasping may not need a precise pose of the robot hand relative to the object, precision grasps and in-hand manipulation require a high level of accuracy [1]. In many cases, the exact model of the robot arm may not be available and forward kinematics is not accurate enough to guide grasping [2]. Vision-based hand pose estimation can alleviate this problem and enable control without an extra step requiring position or image-based visual servoing.

Similar requirements arise when grasping and manipulation tasks are performed by several robots where the relative position of the robots with respect to each other must be known [3]. In this case, one robot can obtain its relative position with respect to another robot by identifying the full pose of the robot body or solely the pose of its hand.

In this paper, we propose an approach for vision-based pose estimation of a robot hand or full-body pose. The method is based on Virtual Visual Servoing that uses RGB-D images together with a CAD model of the robot, to continuously track the pose of a robot with respect to the camera, or between different parts of a robot. The main contributions of this work are:

- The integration of 2-D and 3-D information into the Virtual Visual Servoing framework. Our method, given an approximate initial pose estimate, refines it iteratively to obtain a more precise estimate. We show that the use of 3-D information improves the estimate in comparison to using only 2-D images.
- A method for pose tracking of a robot in joint space given that its configuration is known. This adds the challenge of having to track each of the links of the robot, which places special requirements on rendering for virtual visual servoing. As we will demonstrate, our system allows us to treat each joint in the same way that we treat each of the components of the motion of the robot, thus making it suitable for complex models.

The authors are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology KTH, SE-100 44 Stockholm, Sweden. This work has been supported by EU FP7 grant 288533-RoboHow.cog, Swedish Research Council and Swedish Foundation for Strategic Research. e-mail: {javiergm|ccs|celle|dani}@kth.se

This paper is organized as follows: in Section II we review related work. The proposed methodology is presented in Section III and the results of the experimental evaluation are presented in Section IV. We conclude the paper in Section V.

II. Related Work

In general, it is possible to use any tracking method to retrieve the pose of the manipulator. The existing methods can be divided in two groups: appearance-based (also referred to as global) [6] and feature-based (also referred to as local). These methods differ mostly from each other in the kind of features that are used, the matching algorithm and the optimization method. Appearance-based methods have commonly been used for obtaining the pose of a moving camera [7], [8] or for coarse pose estimation of objects that occupy a substantial portion of the image or are easily segmented [9]–[12]. There are also approaches that rely on the use of fiducial markers [4], [5] that may limit the mobility of the manipulator, due to the requierment of markers being continuously in the visual field of the camera.

The features commonly employed for tracking are corners or edges. These are extracted using some interest point detector [13], [14] and then encoded into a local descriptor [15] to ease the matching of the features with the ones stored in the model. These kinds of features usually work better with textured objects, and can be problematic with robotic manipulators, which usually consist of flat, shiny surfaces which change with illumination. For the optimization part of the method, if the points are correctly matched and detectable in the views with arbitrary precision, three points are enough to solve the problem [16]. In general, more points are needed, and methods exist that are robust in the presence of noise due to incorrect matches or inaccuracy in point detection [17]–[19]. Most of the systems based on these methods use features extracted from 2-D images as input, and are thus highly sensitive to viewpoint changes. Our method, by using a full 3-D CAD model of the tracked object, is more tolerant to viewpoint changes. Some methods, such as [20], also use a CAD model for tracking the object, but can only support simple models, with a few hundreds of polygons, and lack direct support for tracking a complete kinematic chain.

Virtual Visual Servoing (VVS) [21] is an iterative optimization method where given a real image of the object for

which we want to track the pose and an initial estimation of the pose, a model is projected into the image at the estimated pose. Then, features are extracted both in the real and model image, and the difference between the position of the features is used to improve the pose estimation. In our previous work [22], the original method is extended to make use of a full 3D CAD model of the object. The real image contains only color data (no depth information), so the features chosen are the edges detected in the image.

III. Methodology

As stated, our approach is based on VVS where a rendered model of the object is aligned with the object as seen in the current camera image. We now provide the notation, followed by an overview of the method and a detailed description of each component.

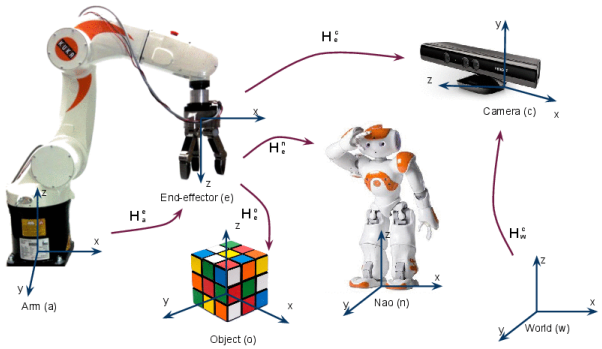


Fig. 1. Coordinate systems and some transformations between them.

The problem we deal with is the estimation of the homogeneous transformation between two coordinate frames, $\boldsymbol{H_o^e}$, relating the corresponding coordinates of a point $\boldsymbol{q_e}$ and $\boldsymbol{q_o}$ through $\boldsymbol{q_e} = \boldsymbol{H_o^e q_o}$, see Fig. 1. Vision-based pose estimation provides us with a means of obtaining the transformation $\boldsymbol{H_x^c}$ between a coordinate frame x and the coordinate frame c of the camera. By obtaining these for e and o, we can obtain the transformation $\boldsymbol{H_o^e} = \boldsymbol{H_e^{c-1} H_o^c}$ between the two original frames. The robot arm is assumed to consist of several links. In general, the relationships between links form a kinematic tree, where the transformation $\boldsymbol{H_{l_i}^a}$ between the root link and link i is $\boldsymbol{H_{l_i}^a} = \boldsymbol{M_{j_i(1)} M_{j_i(2)} \dots M_{j_i(n)}}$ where $\boldsymbol{M_k}$ is the transformation corresponding to joint k and j_i is the sequence of indices of the joints that separate the root link and link i in the kinematic tree. The transformation between the camera and each of the links is then: $\boldsymbol{H_{l_i}^c} = \boldsymbol{H_a^c M_{j_i(1)} M_{j_i(2)} \dots M_{j_i(n)}}$ which is what we wish to estimate. $\boldsymbol{H_a^c}$ is the rigid transformation between the camera and the root of our kinematic tree, composed of a rotation $\boldsymbol{R_a^c}$ and a translation $\boldsymbol{t_a^c}$. If the kinematics and joint configuration of our robot are fully known, $\boldsymbol{M_k}$ will be known, so we only need to estimate the rotation and translation of the base. When the kinematics is known but the joint configuration is not, determining the joint transformation will be equivalent to determining some parameter ϕ_k for the joint (usually an angle). We can then write $\boldsymbol{H_{l_i}^c} = \boldsymbol{H_a^c(R_a^c, t_a^c)} \prod_{s=1}^{n} \boldsymbol{M_{j_i(s)}}(\phi_{j_i(s)})$ where $\boldsymbol{R_a^c}$, $\boldsymbol{t_a^c}$ and $\boldsymbol{\phi} = \{\phi_1, \phi_2, \dots \phi_m\}$ are the parameters to estimate.

A. System overview

The outline of the system is shown in Fig. 2. To achieve the alignment, we can formalize it by either controlling the pose of the *virtual* object or moving the *virtual* camera so that the image perceived by the camera corresponds to the current camera image, denoted as *real* camera image. In this paper, we adopt the first approach achieved through rendering synthetic images by incrementally changing virtual the pose of the robot hand/arm. A rough initial estimate of the pose is given by forward kinematics. Image features are then extracted from the rendered image and matched to the features of the current image of the hand. We define

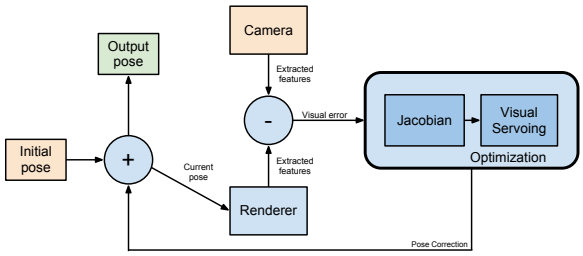


Fig. 2. Outline of the proposed model-based tracking system.

an error vector based on the differences between the image features in the rendered and current real camera image. The error vector is used for the pose estimation process using VVS formalization. Images $I_r(u, v)$ are captured with the Kinect sensor at 30fps. The sensor provides also a depth map $D_r(u, v)$.

B. Synthetic image generation

For rendering, we assume that a CAD model of the robot is available. We developed a new scenegraph engine focusing on rendering offline images and associated maps at a very high speed, using custom shaders with OpenGL. Since our matching is based solely on the edge data, the model is rendered without any texture or lighting, which allows us to obtain more than 1000 fps in modern consumer GPU hardware, for a model containing more than 100000 polygons. For each real image, several iterations of VVS need to be applied before convergence, thus fast rendering is necessary for real-time performance. Typically, 10 to 30 iterations are needed, depending on the initial offset in the pose thus requiring rendering at about 1000 fps when real images are being captured at 30 fps. Our CAD model is broken into N_{link} meshes, one for each link of the manipulator that can move independently. Each point $\boldsymbol{p_{l_i}} = [x_{l_i}, y_{l_i}, z_{l_i}, 1]^T$ in mesh i can be transformed into the camera coordinate system

using

$$\boldsymbol{p_c} = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \boldsymbol{H_w^c H_a^w}(\hat{\boldsymbol{R}}_{\boldsymbol{a}}^{\boldsymbol{w}}, \hat{\boldsymbol{t}}_{\boldsymbol{a}}^{\boldsymbol{w}}) \boldsymbol{H}_{\boldsymbol{l_i}}^{\boldsymbol{a}}(\hat{\boldsymbol{\phi}}) \boldsymbol{p_{l_i}} \qquad (1)$$

where we make it explicit that this transformation depends on the current estimation of the position and rotation of the manipulator and the joint configuration. To project the resulting point into the image plane, we use the projection matrix $\boldsymbol{P}$, which must correspond with the projection matrix for the camera model of the real camera. The point (u, v) in the image can then be obtained as:

$$\boldsymbol{p_p} = \begin{bmatrix} x_p & y_p & z_p & w_p \end{bmatrix}^T = \boldsymbol{P p_c}, \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x_p/w_p \\ y_p/w_p \end{bmatrix} \qquad (2)$$

Using this transformation we generate three different maps. $I_s(u, v)$ contains 1 where a point was rendered and 0 for the background. $D_s(u, v)$ contains the depth of the rendered point. It is also used during the rendering process for occlusion culling. Finally, $T_s(u, v)$ contains the index of the link for the corresponding pixel.

C. Image features

As mentioned before, the rendered and real images are compared using image features. In our previous work [22], we used only edges in the 2D image as a feature. Here, image edges are still used, but they are combined with new features to increase the robustness and accuracy of the system.

1) Image edges: We use a Canny operator for edge detection. We extract edges in both the real and virtual images, and the vector between an edge point in one image and the closest edge point in the other image gives us a directed error vector. For efficiency, this is implemented using the distance transform method: for each real image, a map is created which assigns to each point in the map, the position of the closest edge point. Then, for each edge point in the virtual image, the error vector can be obtained by a simple lookup in that map. Since our method assumes an initial pose estimate, edges should only be matched when their orientations are similar. To enforce that, 8 maps are generated, which record the closest edge within a certain range of orientations. Then, for each edge point in the virtual image, the lookup is performed only in the map which best corresponds to the orientation of the edge point.

2) Image depth: One of the important parts of the system is the choice of appropriate features for the raw depth information. SIFT-like 3-D features, such as the one introduced in [24] are a possibility, but they suffer from the same drawbacks as SIFT for the 2-dimensional case. Robotic surfaces are often flat, and the matching of features is an expensive operation that would need to be performed for every frame. It is also possible to use the depth information directly as a feature. In [25], the depth map is assumed to be smooth, and the difference between the depths of each point in the source and target images is used as a feature. The main drawback of this approach is that it leads to incorrect values in the edges of the object, and is extremely sensitive to small occlusions, such as the ones that can be caused by cables in robotic environments. Also, we do not have depth information for the whole scene, but only for the manipulator, which will usually only cover a small part of the depth map.

The approach we apply is to use the distance from one 3-D point obtained from the virtual image to the closest point obtained in the real image. The 3-D image is actually an edge image, in the sense that each point corresponds to what would be an edge in fronto-parallel 2-D cuts of the scene, so this method has similar advantages to the one we adopted for the 2-D information. We implement it again using a 3-D version of the distance transform, where we create, for the real image, a 3-D map of the distance from each point in space to the nearest extracted point. Then, for each depth point in the virtual image, we just need to perform a lookup for the nearest point in the 3-D map, and we obtain a directed error vector. Another practical advantage of using this method is that it is very similar to the one used for 2-D edges, so it can be easily integrated into our framework.

3) SURF features: The previous features meet the key requirements of speed and work well with textureless objects, but their main drawback is that since each point in one image is compared to the closest point in the other image, the performance degrades when the initial pose is bad. To improve the performance in such cases, we need features that can be robustly matched between the images and we chose to use SURF [26]. Since our CAD models are not textured, we cannot directly detect SURF features in the rendered image. We could generate texture maps for our CAD models, but even then, detecting SURF features for every generated virtual image would be prohibitively expensive. Instead, we enrich our CAD model with pre-detected SURF features. In an offline process, we detect SURF features in different parts of our model, and for each feature we record its 3D position within the CAD model, together with information about the viewpoint, the detection size and the feature descriptor. Then, during the pose estimation loop, SURF features are detected in each captured image, and their descriptors matched against the database of stored features. Matches that are not consistent in terms of viewpoint and detection size are discarded. The distance between the feature as detected in the real image and the projection of the recorded position into the rendered image is then used as the feature to minimize.

D. Visual Servoing

The basic idea behind visual servoing is to create an error vector which is the difference between the desired and measured values for a series of features, and then map this error directly to robot motion. Let $\boldsymbol{s}(t)$ be a vector of feature values which are measured in the image. In our case, it is constructed, at each iteration, with the distances $\boldsymbol{d}$ between the detected points in the real and synthetic images as $\boldsymbol{s}(t) = \begin{bmatrix} d_1, d_2, \ldots, d_n \end{bmatrix}^T$. Then $\dot{\boldsymbol{s}}(t)$ will be the rate of change of these distances with time as $\boldsymbol{H}_{\boldsymbol{a}}^{\boldsymbol{c}}(\boldsymbol{R}_{\boldsymbol{a}}^{\boldsymbol{c}}, \boldsymbol{t}_{\boldsymbol{a}}^{\boldsymbol{c}})$ is updated to improve the fit between real and synthetic images. The change in this transformation can be described

TABLE I

ESTIMATION ERRORS IN THE RETRIEVED POSE FOR KUKA ARM AND NAO ROBOT.

	KUKA arm				**NAO**			
	simulation		real data		simulation		real data	
	2-D features	2-D and 3-D features	2-D features	2-D and 3-D features	2-D features	2-D and 3-D features	2-D features	2-D and 3-D features
Translation error parallel to image plane	11.3 mm	9.8 mm	15.8 mm	12.1 mm	10.2 mm	9.7 mm	17.1 mm	11.7 mm
Translation error perpendicular to image plane	40.1 mm	9.2 mm	46.3 mm	9.7 mm	30.7 mm	9.9 mm	39.3 mm	9.6 mm
Rotation error	1.01 °	0.63 °	1.43 °	0.93 °	1.23 °	0.79 °	1.17 °	1.08 °

by a translational velocity $\boldsymbol{T}(t) = [T_x(t), T_y(t), T_z(t)]^T$ and a rotational velocity $\boldsymbol{\Omega}(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$, which form a velocity screw: $\dot{\boldsymbol{r}}(t) = \left[T_x, T_y, T_z, \omega_x, \omega_y, \omega_z\right]^T$. We can then define the image jacobian or interaction at a certain instant as $\boldsymbol{J}$ so that $\dot{\boldsymbol{s}} = \boldsymbol{J}\dot{\boldsymbol{r}}$ where

$$\boldsymbol{J} = \left[\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{r}}\right] = \begin{bmatrix} \frac{\partial d_1}{\partial T_x} & \frac{\partial d_1}{\partial T_y} & \frac{\partial d_1}{\partial T_z} & \frac{\partial d_1}{\partial \omega_x} & \frac{\partial d_1}{\partial \omega_y} & \frac{\partial d_1}{\partial \omega_z} \\ \frac{\partial d_2}{\partial T_x} & \frac{\partial d_2}{\partial T_y} & \frac{\partial d_2}{\partial T_z} & \frac{\partial d_2}{\partial \omega_x} & \frac{\partial d_2}{\partial \omega_y} & \frac{\partial d_2}{\partial \omega_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_n}{\partial T_x} & \frac{\partial d_n}{\partial T_y} & \frac{\partial d_n}{\partial T_z} & \frac{\partial d_n}{\partial \omega_x} & \frac{\partial d_n}{\partial \omega_y} & \frac{\partial d_n}{\partial \omega_z} \end{bmatrix} \quad (3)$$

which relates the motion of the (virtual) manipulator to the variation in the features. The method used to calculate the jacobian is described in detail below.

However, what we need to be able to correct our pose estimation is the opposite, that is, we need to compute $\dot{\boldsymbol{r}}(t)$ given $\dot{\boldsymbol{s}}(t)$. When $\boldsymbol{J}$ is square and nonsingular, it is invertible, and then $\dot{\boldsymbol{r}} = \boldsymbol{J}^{-1}\dot{\boldsymbol{s}}$. This is not generally the case, so we have to compute a least squares solution, which is given by $\dot{\boldsymbol{r}} = \boldsymbol{J}^{+}\dot{\boldsymbol{s}}$ where $\boldsymbol{J}^{+}$ is the pseudoinverse of $\boldsymbol{J}$ calculated as $\boldsymbol{J}^{+} = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T$. The goal for our task is for all the edges in our synthetic image to match edges in the real image, so the target value for each feature is 0, and we can define the error function as $\boldsymbol{e}(\boldsymbol{s}) = \dot{\boldsymbol{s}} - \boldsymbol{0}$ which leads us to the simple proportional control law $\dot{\boldsymbol{r}} = -K\boldsymbol{J}^{+}\dot{\boldsymbol{s}}$ where K is the gain parameter.

E. Estimation of the jacobian

To estimate the jacobian we need to calculate the partial derivatives of the feature values d_i with respect to each of the components of the motion we are estimating ($\boldsymbol{R_a^c}$, $\boldsymbol{t_a^c}$ and ϕ). When features are the position of points or lines, it is possible to find analytical solutions for the derivatives. Here, however, the features are the distances from the edges of the synthetic image to the closest edge in the real image, so we approximate the derivative by calculating how a change in the motion component affects the value of the feature.

Each of the feature values d_i is the distance between a point $\boldsymbol{p_i^s(u,v)}$ in the synthetic image and the corresponding point $\boldsymbol{p_i^r(u,v)}$ in the real image. We want to find the point $\boldsymbol{p_i^{s\prime}(u,v)}$ which results from applying the small change in the motion component to $\boldsymbol{p_i^s(u,v)}$. We can use the depth map $D_s(u,v)$ to find the corresponding 3D point in camera coordinates, and then use the inverse of the matrix that we used to render the point from the model to find the point $\boldsymbol{p_i^m(x,y,z)}$ in the coordinate system of the model. Different points in the image will correspond to different links in the robot, but we can obtain the link for each point, and thus its corresponding projection matrix from map $T_s(u,v)$.

Once we have $\boldsymbol{p_i^m(x,y,z)}$, we can reproject it using the new transformation matrix which would result from applying the small change in motion component, obtaining, as we wanted, $\boldsymbol{p_i^{s\prime}(u,v)}$. We then compute the new distance d_i' to the corresponding point in the real image, and we can estimate the derivative as $(d_i' - d_i)/\epsilon$, where ϵ is the change in motion component.

IV. EXPERIMENTAL EVALUATION

We test the performance of the method with respect to the choice of 3D features. We then give a more extensive evaluation of the method's performance in different situations, demonstrating hand-eye calibration or robot pose estimation.

A. Accuracy evaluation and comparison to previous method

We first evaluated the accuracy in the pose estimation in tracking two robots: A KUKA industrial arm and a NAO humanoid robot. We performed tests both with imagery from a simulator and with real-world data obtained from a Kinect camera. The results, which include a comparison with our previous method which used only 2-D information are summarized in Table III-D. Each value is the average error over 1000 runs. We used 5 different joint configurations for each robot and 10 different initial estimates for the pose, giving the total of 50 starting conditions. Examples of initial and final position for a run are shown in Figures 3 and 4.

To evaluate the error in the real-world experiments, we needed ground truth. We chose to compare the results to how a human would manually align the input point cloud with the rendered CAD model, using the same information available to the robot. For the position error, we distinguish between errors that are parallel or perpendicular to the image plane, and we observe that the errors in the estimation of the depth of the object are greatly reduced.

B. Convergence of the method

To evaluate the robustness of the method, we estimate the maximum error in the initial pose estimation for which the method will still converge, using the KUKA industrial arm and real-world imagery. In this set of experiments,

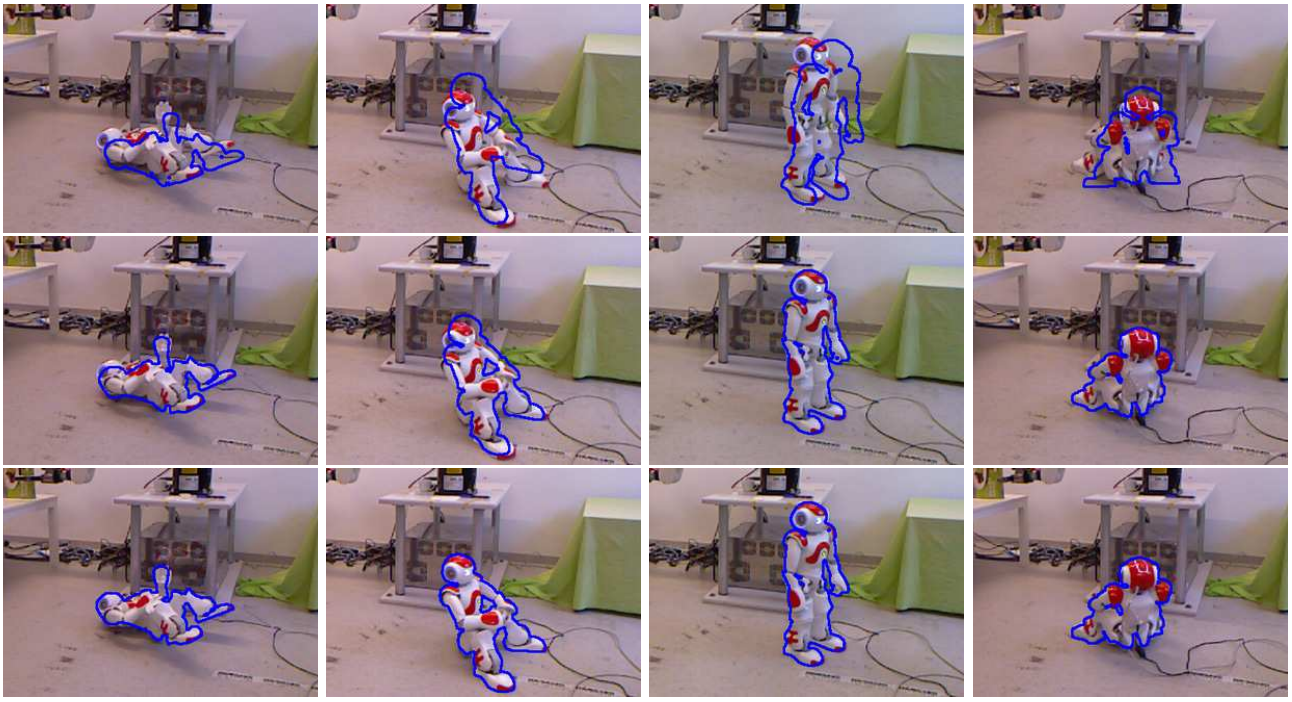


Fig. 3. A few examples of the initial (upper row) and final poses (lower row) for a Nao robot in several different configurations. The blue outline represents the current estimation of the pose. Best viewed in color.

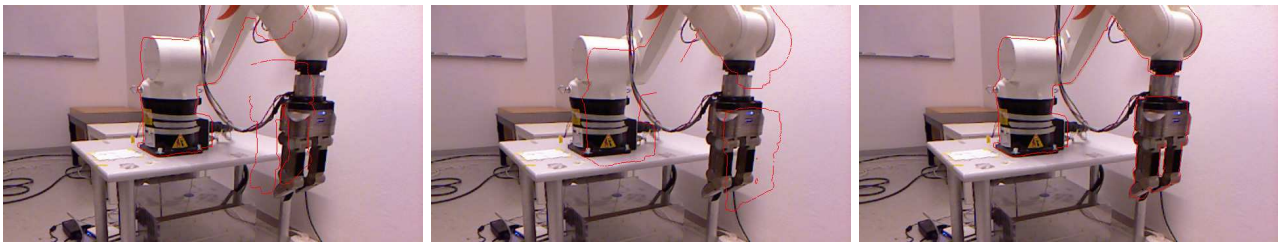


Fig. 4. Initial poses with (a) errors in the joint positions (b) errors in the transformation for the whole manipulator. (c) Converged result. Red outline represents the current estimation. Best viewed in color.

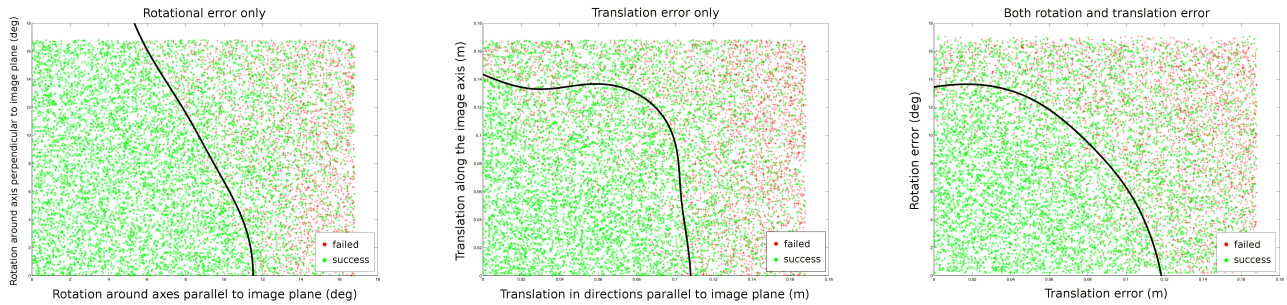


Fig. 5. Convergence results for (a) only rotational error (b) only translational error (c) both rotational and translational error. Best viewed in color.

we also assume that the joint configuration is known. We performed a total of 30000 runs of the method, using five different joint configurations for the manipulator. The results for different kinds of errors, including a decision boundary for convergence can be seen in Figure 5.

We can observe that for translational errors of less than 10 cm and rotational errors of less than 10 degrees, the method converges with high probability. Also, we can see that translational errors along the axis perpendicular to the image plane and rotational errors around that same axis, a

larger error is tolerated.

C. Estimation of joint configuration

Until now, we have assumed a known joint configuration. While this is the case in our system, it is not true for many robotic manipulators. In the following set of experiments, we assume that the transformation with respect to the base of the manipulator is known, but there is some error in the initial estimate of the joint configuration. Having the real values as provided by our system allows us to compare the results of our method with the true values.

We ran our method 10000 times for the KUKA arm using real-world images, with 5 different target (real) joint configurations, and each time introducing an error of between -5 and 5 degrees to each of the 6 joints of our arm. A total of 91% of the runs converged, and the average mean-square-error over the joints for each run was 0.83 degrees.

V. Conclusion and Future work

We have proposed an approach for vision based pose estimation of a robot hand or full body pose. The method is based on virtual visual servoing using a CAD model of the robot. The method combines 2-D image features with depth features. The method can be applied to estimate either the pose or the full configuration of a robot. We presented experimental, demonstrating the performance of the approach on both a mobile humanoid robot and a stationary manipulator.

Our experiments show that considering three-dimensional features which can be easily obtained from RGB-D images significantly improves performance when tracking robots, especially with respect to the perception of the distance from the camera to the robot. We have successfully applied the method to the tracking of a walking humanoid, as can be seen in the accompanying video. We also showed that the method can be used to refine the estimation for the joints of a robotic manipulator, where limitations in the hardware introduce uncertainties.

However, when combining both errors in the transformation for the base and in the joint configuration, the current method is stable only for limited ranges of errors. We need further studies on the relative benefits of 3D features depending on how large these errors are. Preliminary tests show that the 3D features used are complimentory. Whereas SURF features are most valuable for large errors, edges are important when errors are small. This leads to the conclusion that a system could benefit from varying the contribution of different features depending on how far you are from converging. Our plan is to continue in this direction, and gradually increase the radius of convergence, while keeping the same high accuracy.

References

[1] T. Feix, H. Schmiedmayer, J. Romero, and D. Kragic, "A comprehensive grasp taxonomy," in *In Robotics, Science and Systems: Workshop on understanding the human hand for advancing robotic manipulation*, 2009.

[2] N. Vahrenkamp, S. Wieland, P. Azad, D. Gonzalez, T. Asfour, and R. Dillmann, "Visual servoing for humanoid grasping and manipulation tasks," in *IEEE-RAS International Conference on Humanoids*. IEEE, 2008, pp. 406–412.

[3] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation - a survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[4] H. Kato and M. Billinghurst, "Developing AR applications with ARToolkit," in *ISMAR*. IEEE Computer Society, 2004, p. 305.

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

[6] M. Irani and P. Anandan, "About direct methods," *Vision Algorithms: Theory and Practice*, pp. 267–277, 2000.

[7] M. Meilland, A. Comport, and P. Rives, "Real-time dense visual tracking under large lighting variations," in *British Machine Vision Conference, University of Dundee*, vol. 29, 2011.

[8] G. Caron, A. Dame *et al.*, "L'information mutuelle pour l'estimation visuelle directe de pose," in *Actes de la conférence RFIA 2012*, 2012.

[9] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2155–2162.

[10] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 585–592.

[11] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition," in *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, 2011.

[12] M. Arie-Nachimson and R. Basri, "Constructing implicit 3d shape models for pose estimation," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 1341–1348.

[13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[14] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Computer Vision—ECCV 2002*, pp. 128–142, 2002.

[15] ——, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[16] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle, "Analysis and solutions of the three point perspective pose estimation problem," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1991, pp. 592–598.

[17] R. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

[18] D. Oberkampf, D. DeMenthon, and L. Davis, "Iterative pose estimation using coplanar points," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1993, pp. 626–627.

[19] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, 2003.

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

[21] A. I. Comport, É. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Trans. Vis. Comput. Graph*, vol. 12, no. 4, pp. 615–628, 2006.

[22] X. Gratal, J. Romero, and D. Kragic, "Virtual visual servoing for real-time robot pose estimation," in *Proceedings of the 18th IFAC world congress*, 2011.

[23] D. Kragic and V. Kyrki, "Initialization and system modeling in 3-d pose tracking," in *IEEE International Conference on Pattern Recognition*, Hong Kong, 2006, pp. 643–646.

[24] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.

[25] C. Teuliere and E. Marchand, "Direct 3d servoing using dense depth maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1741–1746.

[26] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[27] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

Predicting Slippage and Learning Manipulation Affordances through Gaussian Process Regression

Francisco E. Viña B., Yasemin Bekiroglu, Christian Smith, Yiannis Karayiannidis, Danica Kragic

***Abstract*— Object grasping is commonly followed by some form of object manipulation – either when using the grasped object as a tool or actively changing its position in the hand through in-hand manipulation to afford further interaction. In this process, slippage may occur due to inappropriate contact forces, various types of noise and/or due to the unexpected interaction or collision with the environment.**

In this paper, we study the problem of identifying continuous bounds on the forces and torques that can be applied on a grasped object before slippage occurs. We model the problem as kinesthetic rather than cutaneous learning given that the measurements originate from a wrist mounted force-torque sensor. Given the continuous output, this regression problem is solved using a Gaussian Process approach.

We demonstrate a dual armed humanoid robot that can autonomously learn force and torque bounds and use these to execute actions on objects such as sliding and pushing. We show that the model can be used not only for the detection of maximum allowable forces and torques but also for potentially identifying what types of tasks, denoted as *manipulation affordances*, a specific grasp configuration allows. The latter can then be used to either avoid specific motions or as a simple step of achieving in-hand manipulation of objects through interaction with the environment.

I. Introduction

Interaction with and manipulation of objects are essential abilities of robots operating in realistic environments. As humans, robots may need to grasp objects for simple tasks such as moving them from one position to another. More complex tasks, such as using objects as tools, requires a more advanced ability of manipulating an object in the hand so to achieve a suitable grasp configuration. In this process of achieving and loosing contacts with the object in the hand, events such as slippage commonly occur. The knowledge of contacts and slippage provides important information about the status of the task one is executing.

For both humans and robots, sense of touch is paramount for safe and flexible interaction with objects and the environment. As reviewed in [1], components of tactile perception in humans depend on the sensory inputs within muscles, tendons and joints (kinesthetic) and stimulus mediated by receptors in the skin (cutaneous). Most of the research in robotic tactile sensing addressed the problem of finger-object interactions and grasp stability assessment. If the contact locations as well as the friction coefficients of the contacting surfaces are known, the problem can be formulated in terms of the Grasp Wrench Space (GWS) [2], [3]. However, it is difficult to construct the GWS in practice since it requires the exact values of those parameters.

The authors are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology KTH, SE-100 44 Stockholm, Sweden. e-mail: {fevb|yaseminb|ccs|yiankar|dani}@kth.se

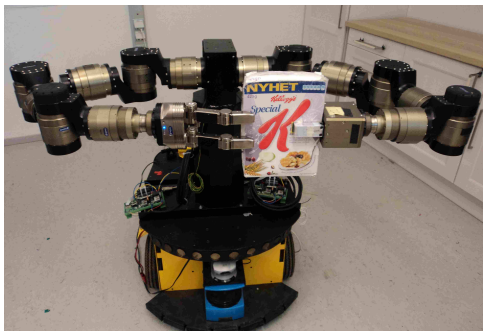


Fig. 1 : A dual arm robot setup for estimating maximal allowable forces and torques for a grasp.

Besides planning stable grasps, the robot should also acquire knowledge of the maximum forces and torques that can be applied on the object before slippage occurs. Various methods have been proposed for detecting slippage [1], [4]–[6]. Apart from addressing the problem at the signal processing level in terms of cutaneous tactile sensing, general machine learning methods have proven adequate for analysis in cases where noise and imperfect models are inherent to the problem, [7], [8].

Our work follows the direction of using kinesthetic sensing for slip detection in combination with machine learning techniques. Autonomous learning and a physical model of the friction forces are used to estimate the maximum static friction forces and torques on objects the robot is interacting with. We approach the problem through Gaussian Process regression, resulting in a model that can predict forces and torques that a grasp can tolerate before the held object starts slipping. As such, the model can also be used to identify the affordances of a specific grasp such as, for example, what type of in-hand rotation can be applied to an object while still keeping the object in the hand.

The learned bounds can be used as constraints at the control level to avoid certain motions and thus prevent slippage of the grasped object while executing the task. In addition, the approach also identifies in which directions the object might translate or rotate in the hand and thus be exploited in tool use and in-hand manipulation to actively change the pose of the object in the hand – either through specific motion or interaction with the environment. This is also commonly done by humans, for example prior to putting a key in a keyhole we may change its orientation between the fingers by pushing the key toward a surface.

Thus, differently from commonly addressed *grasp affordances* [9], we facilitate the system to identify *manipulation affordances*. Our method uses force-torque and proprioceptive feedback different from commonly used tactile or skin sensors which in practice can be fragile and easily damaged. However, when possible, the cutaneous and kinesthetic methods can be integrated resulting in a more biologically inspired approach [1]. Our approach also takes advantage of the dual arm capabilities of humanoid robots since the training actions can be executed autonomously through dual arm manipulation procedures. Fig. 1 shows our dual-arm robot as an example of a platform that can be used to implement the method we propose in this paper.

The paper is organized as follows: Section II presents the related work, Section III our learning framework, including the friction model and the use of Gaussian Process regression while in Section IV we proceed to describe how our system learns manipulation affordances from doing regression on the static friction. Finally, we provide our experimental results in Section V as well as the conclusions, discussion on the results and future directions in Section VI.

II. Related Work

Early works studying the physics of robotic grasping and contact between rigid bodies are reviewed in [3]. The review addressed the basic closure properties of grasps, force and form closure, which describe the equilibrium conditions of an object grasped by a robotic hand by assuming frictional and frictionless point contacts respectively. Given that friction forces play a central role in robotic grasping, some of the works reported in the literature have focused on studying their properties [5], [10]. These studies cover not only the translational Coulomb friction, but also the rotational friction. Moreover, by combining different sensor modalities (tactile and force-torque) it is shown in [5] that it is possible to detect and control both translational and rotational slippage.

Besides modeling the physics of grasping and the friction forces, quantifying the quality of grasps in terms of the capability to counteract external disturbances has been one of the main research questions in the grasping community. In order to plan stable grasps with robotic hands, many grasp planners have been proposed in the literature which optimize these quality measures [2], [11], [12]. These planners are constructed in terms of approximations of wrench spaces or heuristic algorithms that consider a subset of a wrench space.

The main drawback of these methods is that these require precise 3D models of the object as well as prior knowledge of the friction coefficient and the location of the contact points of the robot's hand. To cope with this problem, [13] proposes a set of manipulation actions to estimate properties such as weight, stiffness and friction in order to set appropriate grasping forces.

In order to overcome the uncertainties and problems with modeling errors in grasping, learning approaches have also been proposed. Example works of [7], [8], [14] consider learning of grasp stability and grasp affordances. Our previous work on grasp stability assessment performs learning mainly through tactile (cutaneous), proprioceptive and visual feedback in order to predict the stability of the grasp prior to lifting and manipulating the object [8], [14]. In [7] the proposed system learns grasp affordances which are defined as hand-object relative poses that lead to successful grasps on a particular object. These affordance densities are learned through exploration and visual features. The main strength of these learning approaches originates from the fact that these do not require prior knowledge of physical contact parameters as the system is trained using supervised learning without explicitly modeling the physics of grasping.

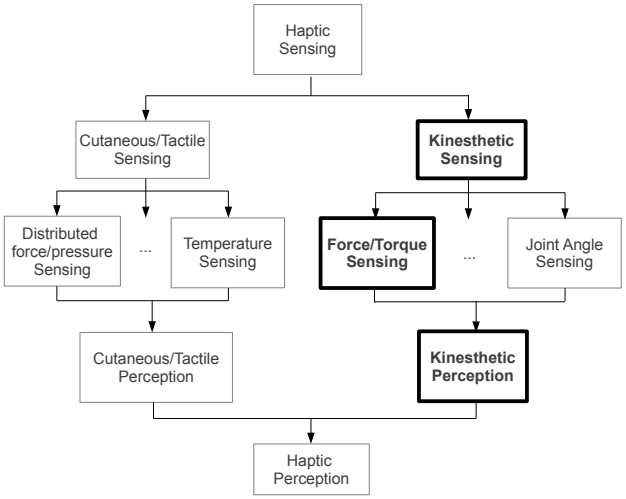


Fig. 2 : Cutaneous and kinesthetic components of haptic sensing and perception [1], [15]. Highlighted in bold are the kinesthetic components which we consider in our approach.

Our work makes use of the physics models of friction described in the seminal work of [5], [10]. However, instead of employing geometrical, analytical or signal processing based approaches [2], [4], [5], [11], [12] we follow a kinesthetic learning approach for predicting slippage. In this sense, our work follows more closely approaches in which the robot first interacts with objects and assesses their contact and friction properties prior to executing tasks [13]. Our method also follows the motivation behind learning based approaches in order to deal with the issue of modeling errors and uncertainties in grasping [7], [8], [14].

Within the broader scope of *haptic* sensing, which consists of both cutaneous and kinesthetic sensing as shown in Fig. 2, our approach falls under the subcategory of kinesthetic sensing and perception while most of the related work discussed so far including our own work on grasp stability assessment cover mostly the domain of cutaneous/tactile sensing [4], [6], [8], [14].

III. Physics and Learning Model

The main objective of our system is learning the maximum static friction forces and torques for various grasp configurations through force-torque sensing. In this section we present

the modeling aspects of our framework, beginning with a description of the friction model used and the selection of input features for training. We finalize the section with a brief overview of Gaussian Process regression and explain how we apply it within our work.

A. Friction Model

According to the Coulomb friction model, when an external force is applied parallel to the surface of contact between two bodies, there is a reaction friction force f_f which relates to the normal force f_n according to the following inequality

$$f_f \leq \mu_s f_n \tag{1}$$

where μ_s is the static coefficient of friction. This equation holds until the external force exceeds the maximum static friction force. The object then starts slippping when Eq. (1) becomes an equality. From this point, a dynamic friction force with a lower friction coefficient starts acting on the object as depicted in Fig. 3. The peak of this curve corresponds to the maximum static friction force f_{slip} given by

$$f_{slip} = \mu_s f_n \tag{2}$$

The static torsional friction typically displays a nonlinear behavior given by

$$\tau_{slip} = \beta_s f_n^{4/3} \tag{3}$$

where β_s depends on geometric and elasticity factors of the contact [5]. However, slippage still occurs at the point in which the friction torque reaches its maximum value, which we denote as τ_{slip}.

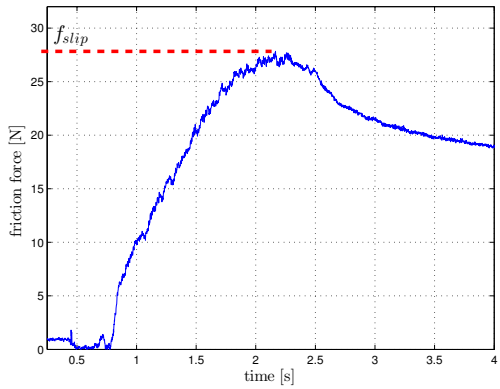


Fig. 3 : Translational friction force exerted on an object held in a robot hand. The peak of the signal, f_{slip} denotes the maximum static friction force at which the object begins to slip.

In order to achieve a more general physical model for prediction, we take into consideration the effect of both rotational and translational friction forces as discussed in [5], [16]. When an object is subject to both rotational and translational shears, the translational and rotational friction components become correlated as shown in Fig. 4. The curve $f_t = h(\tau_n)$, where f_t is the component of the force tangent to the contacting surfaces and τ_n the component of the torque in the normal direction, represents the boundary at which the object starts slipping due to the loads exerted on the object. If the tangential force f_t applied on the object is above the curve for a given applied torque τ_n, then the object will slip and the grasp is thus unstable.

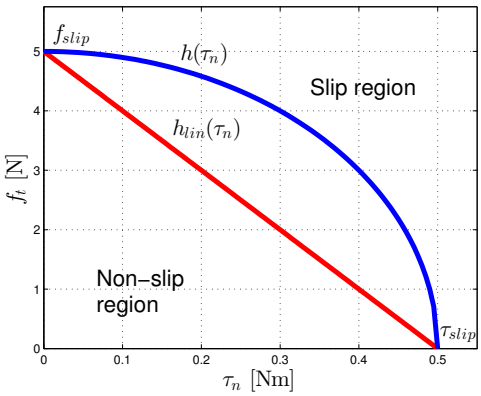


Fig. 4 : Slippage boundaries: $f_t = h(\tau_n)$ represents the boundary for slippage of objects under combined translational and rotational shear while $f_t = h_{lin}(\tau_n)$ represents a linear approximation of h as proposed in [5].

A number of mathematical approximations have been formulated in the literature to describe this slippage boundary. We will use the linear approximation described in [5] that defines a conservative bound on the magnitude of the forces and torques that cause slippage on an object. This linear bound is denoted by $f_t(\tau_n) = h_{lin}(\tau_n)$ in Fig. 4 and can be expressed using the following equation:

$$\frac{f_t}{\mu_s} + \frac{\tau_n}{\beta_s} = f_n \tag{4}$$

B. Learning Framework

Our goal is to learn the mapping between a set of input features (X) and the resulting maximum friction forces and torques (Y), which is a regression problem due to the continuous outputs. While there are several types of regression techniques that could be used within our framework, we have chosen Gaussian Process (GP) regression which can capture the nonlinearity in the data and provide estimates for uncertainty in the predictions.

1) Gaussian Processes: Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ with n observations where $\mathbf{x}_i \in \Re^N$ and $y_i \in \Re$ is a scalar output, regression analysis aims at learning a model for the relationship $y = f(\mathbf{x}) + \varepsilon$ which is composed of a latent function of the input and a noise component ε. As a result of this learning, given a new input $\mathbf{x}^*$, the aim is to obtain the predictive distribution for y^*.

A GP [17] defines a distribution over functions and is parametrized by a mean and a covariance function as

$$GP \sim (m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{5}$$

The mean function is assumed to be zero. The covariance function expresses how similar two outputs, $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$

are given the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$. Our covariance function is based on the squared exponential, which is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp[-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2l^2}] + \sigma_n^2 \delta(\mathbf{x}_i, \mathbf{x}_j). \quad (6)$$

The hyperparameters of the covariance function, (σ_f, σ_n, l), are optimized based on $\mathcal{D}$, where σ_f denotes the signal variance, σ_n is for the noise variance and l is the length-scale which determines how relevant an input is, i.e., if l has a large value the covariance will be independent of that input.

We are interested in the conditional probability $p(y^*|\mathcal{D}, \mathbf{x}^*)$ as we want to find how likely is a certain prediction for y^*, given the data and the new input. Based on a trained GP model, the estimate for y^* is given by the mean value at the test point with the confidence being the variance. The interested reader can refer to the literature [17] for additional details on Gaussian Processes.

2) Feature Selection: As an input to the regressor, we need a set of informative features X, that can reliably represent the behavior of the maximum static friction forces and torques. In our case, we have selected the x component of the hand H pose with respect to the object O as shown in Fig. 5

$$X = \begin{bmatrix} {}^{O}x_H \end{bmatrix} \quad (7)$$

We have selected this feature for illustration purposes, yet more features can easily be incorporated into the system, such as for example the joint angles of the fingers and their grasping force which can modify the friction forces present in a grasp. If more features are incorporated into the system, a preprocessing stage with dimensionality reduction would be necessary [18].

Fig. 5 : Grasp preshape used for training on the maximum static friction forces and torques, with the corresponding reference frames of the hand and the object used for training.

The outputs Y of the regression system are the maximum static friction force and torque

$$Y = \begin{bmatrix} f_{slip} \\ \tau_{slip} \end{bmatrix} \quad (8)$$

which can be measured through force-torque sensors by interacting with the object. We isolate the components of Y and train two GPs, one for the translational friction f_{slip} and one for the rotational friction τ_{slip}. In our case, we learn friction forces f_{slip} in the $y_H - z_H$ plane and friction torques τ_{slip} around the x_H axis of the tip of the hand reference frame as shown in Fig. 5, given that these are the directions in which the object can move within the hand. Forces and torques around the remaining axes are trivial to learn since they will be constrained by the operational safety limits of the hand, given the geometry of the grasp.

IV. Towards Learning Manipulation Affordances

Once the robot has interacted with an object and learned the maximum friction forces $Y = [f_{slip}, \tau_{slip}]^T$ for a range of grasp configurations, it can use this information to infer what type of motions the object can withstand given the current grasp. The details of the training data generation for learning are provided in the next section.

For a given wrench $\boldsymbol{w}^*$ measured by the robot while executing a task, the robot can detect how close the object is to slipping according to the model discussed in Section III-A. In order for the object to remain fixed in the robot's hand the measured force should lie below the torque dependent slippage boundary $h(\tau)$

$$f_t^* < h(\tau_n^*) \quad (9)$$

where f_t^* and τ_n^* are the tangential force and normal torque components of the wrench measured by the robot.

In the training stage we isolate the translational and rotational components of the friction and thus we can approximate $h(\tau_n)$ linearly with $h_{lin}(\tau_n)$ by joining the end points $(f_t, \tau_n) = (f_{slip}, 0)$ and $(f_t, \tau_n) = (0, \tau_{slip})$. In the case of a linear approximation the following condition ensures a stable grasp in terms of zero relative motion between the object and the hand ${}^{H}\mathbf{v}_O = \mathbf{0}$:

$$\begin{aligned} f_t^* &< h_{lin}(\tau_n^*) \\ f_t^* &< -\tfrac{f_{slip}}{\tau_{slip}} \tau_n^* + f_{slip} \end{aligned} \quad (10)$$

Thus, our approach makes it possible to identify stable grasps through identification of forces and torques that can be applied on an object before slippage occurs. In a broader sense, the methodology also identifies directions of motion constraints – that is, in which directions the object is more likely to translate or rotate.

In the case of the grasp studied in this work, see Fig. 5, the model would inform that the object can translate in the $y_H - z_H$ plane and rotate around the x_H axis. Moreover, if a large torque is detected around the x_H axis with relatively low forces in the $y_H - z_H$ plane then we can expect the object to rotate around the fingertips rather than translate once the force-torque measurements reach the slippage boundary of Eq. (4).

This knowledge is necessary for manipulation tasks where a predicted slippage of the object may be facilitated to complete a task. An example scenario is shown in Fig. 6, in which the robot exploits the rotational slippage to pour the contents of the cereal box into the bowl by letting the box rest against an edge of the bowl and allowing it to rotate slightly in the hand while the manipulator moves upwards.

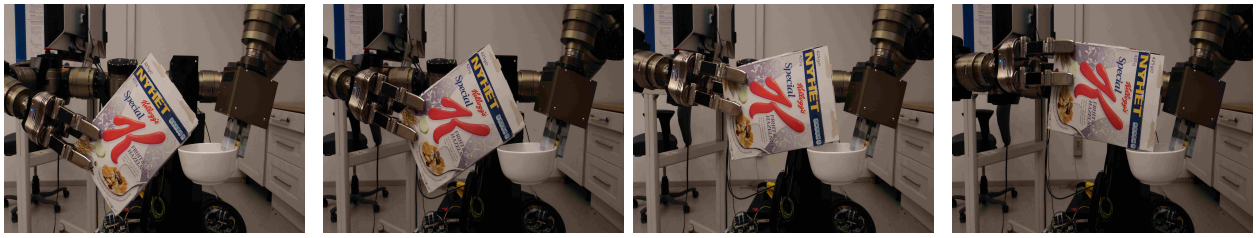


Fig. 6 : Example scenario of a pouring task with rotational slippage.

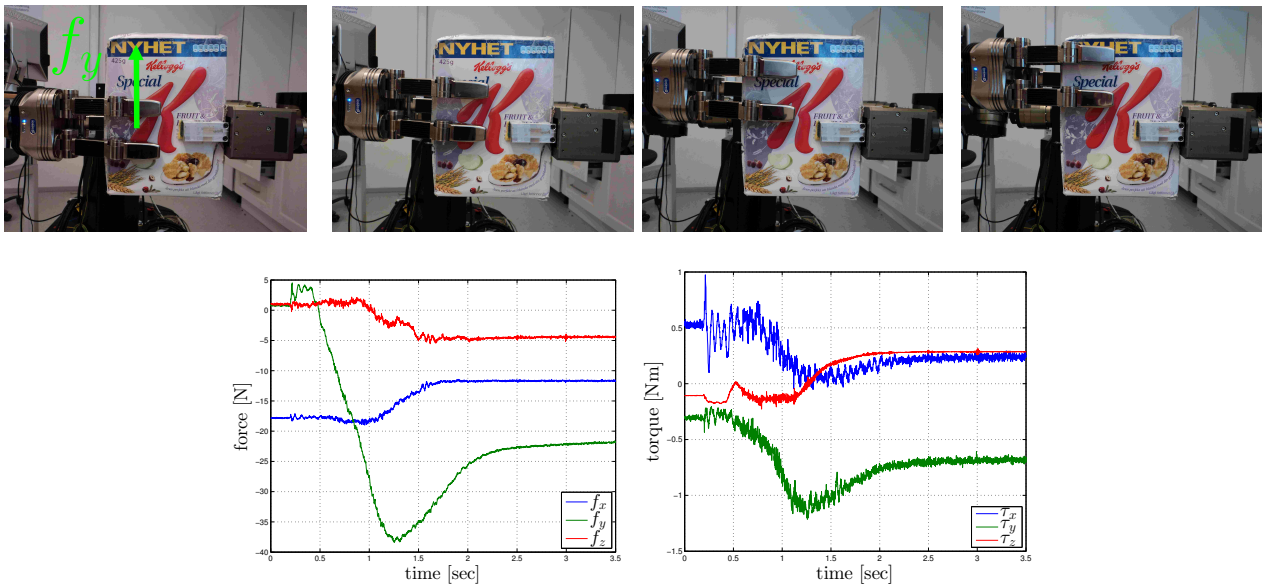


Fig. 7 : Sliding action for training on the maximum static linear friction f_{slip} and its corresponding force and torque profiles.

V. Experimental evaluation

Our experimental setup consists of a dual arm robot as shown in Fig. 1. Each manipulator has 7 DOF and these are equipped with ATI Mini45 6-DOF force/torque sensors mounted at the wrists and they are sampled at a 650 Hz frequency. We start by describing the training data collection process.

A. Training Data Collection

For collecting training data autonomously with the robot we use three dual arm manipulation procedures: one sliding action for measuring the maximum static linear friction f_{slip} and the other two are a rotational motion and pushing action for measuring the rotational friction τ_{slip}.

Fig. 7 shows an illustration of the sliding action along with the forces and torques measured during the execution. In this case the robot holds the object firmly with the parallel gripper shown on the right while the hand on the left, which is the one we train for, slides up in the y_H direction of the hand. The y-component of the force signal f_y measured in the force-torque sensor of the arm is then similar to the one shown in Fig. 3, and f_{slip} is obtained from the peak of the signal.

For obtaining training data for the maximum static friction torque τ_{slip}, we used the pushing action shown in Fig. 8. This action is performed by grasping the object with the hand we train for, while the parallel gripper shown on the right pushes the object on a corner so that the object rotates around the x_H axis of the tip of the robotic hand. We selected this action given that we expect collisions with the environment to be a source of rotational slippage when the robot performs tasks with the object.

For verification purposes we also trained a separate GP for τ_{slip} by applying a different type of training action as shown in Fig. 9. This training action consists of performing a rotational motion with the grasping hand while the object is kept on a fixed grasp with the parallel gripper shown on the right. Even though in this case we also train for τ_{slip} as with the pushing action, we can expect different outcomes from the learning given that each training action represents a different kind of interaction with the environment. The pushing action gives τ_{slip} for tasks in which the object is grasped by the robot's hand and it collides with the environment while being grasped by the robot hand, whereas the rotational motion models a task in which the object is fixed with respect to the environment and the robot's hand rotates around the object.

B. Experimental results

We collected 14 training examples for the friction force and 10 training examples for the torque by varying the relative pose between the robot hand and the manipulated object along one dimension as described in Section V-A. To learn the Gaussian Processes and obtain the hyperparameters we

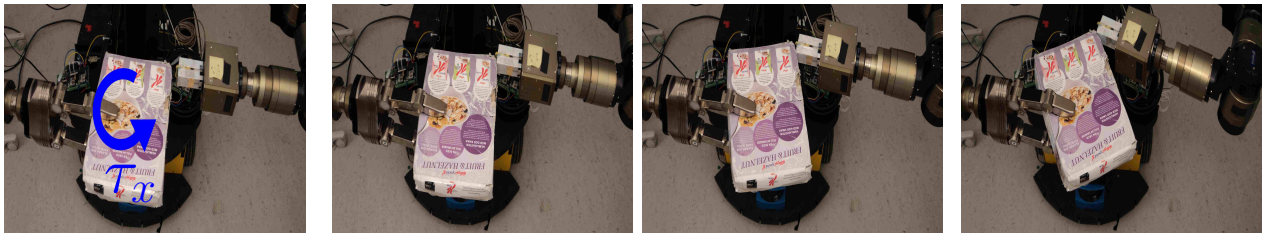


Fig. 8 : Pushing action for training on the maximum static rotational friction τ_{slip}.

Fig. 9 : Rotational motion for training on the maximum static rotational friction τ_{slip}.

used Rasmussen and Nickisch's Gaussian Process Regression and Classification Toolbox [17]. The hyperparameters were calculated by maximizing a Gaussian likelihood function.

Fig. 10 shows the resulting learned Gaussian Process for f_{slip}. This plot shows the mean function of the learned GP (solid blue line) which follows the training points, along with the two standard deviation confidence bounds (dashed red lines) enveloping it. Given this result, we take the lower confidence bound as stability boundary for f_{slip} given that the Gaussian Process predicts that 95% of the points of the process will lie above this boundary.

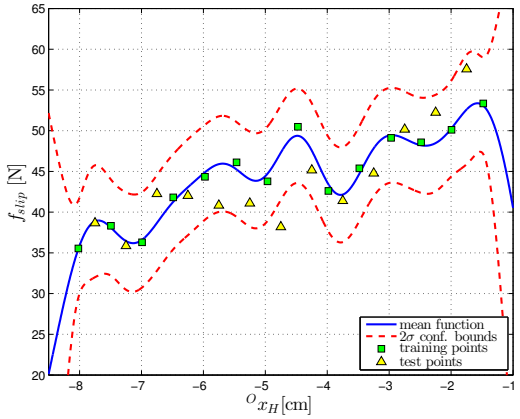


Fig. 10 : Learned GP of f_{slip} with two-standard deviation confidence bounds. The solid blue line is the mean function of the GP while the dashed red lines are the confidence bounds. The green square markers correspond to the training data, while the yellow triangular markers correspond to the test set.

For testing and validating the learned GP, we manually pushed the object while it was being grasped by the robot in different configurations compared to the ones used for training. Fig. 10 confirms that the sliding action performed on the object is valid for training f_{slip} as most of the test points lie above the lower confidence bound of the Gaussian Process.

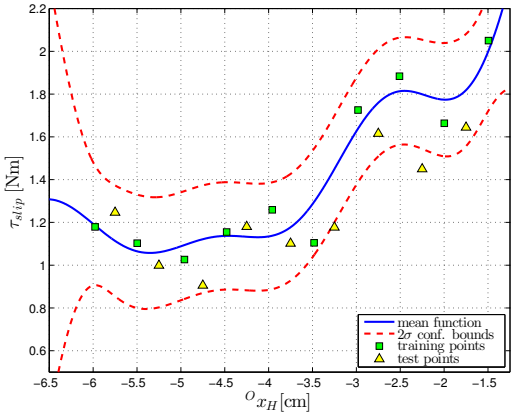


Fig. 11 : Learned GP of τ_{slip} trained by using the pushing action shown in Fig. 8.

Fig. 11 shows the learned Gaussian Process for τ_{slip} when using the pushing action. Once again, we manually pushed the object while it was grasped by the robot in order to collect the test points shown in the figure. These test points show that the pushing action and the learned Gaussian Process succeeded in capturing the behavior of τ_{slip} with respect to the object to hand relative pose.

Fig. 12 shows the result of learning τ_{slip} by using the rotational motion, while we collected test points by manually pushing the object as in the previous case. The clear offset between the learned GP and the test points shows that the training and testing actions are not anymore physically consistent. In the case of the rotational training motion, the interaction between the active robot hand and the object involves both forces and torques, while pushing actions, performed either by the robot hand or manually by ourselves for testing, exert only forces on the object. This result can thus be used to inform the system that the action is not proceeding according to the model and provide the basis for replanning. This is something we plan to adress in the subsequent work.

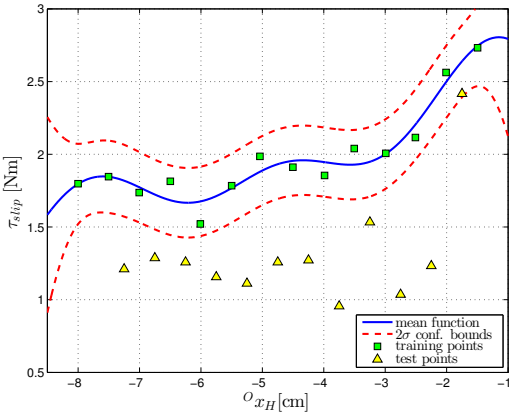


Fig. 12 : Learned GP of τ_{slip} with two-standard deviation confidence bounds trained with the rotational motion shown in Fig. 9.

VI. Conclusions and Future Work

In this work we have presented a learning framework for prediction of slippage of grasps through kinesthetic perception which provides a basis for learning manipulation affordances. Our method uses Gaussian Process regression and the training is performed by isolating the translational and rotational components of the friction. The novelty of the approach lies on using a machine learning approach together with a physical model of the friction to determine continuous bounds on the forces and torques that a grasped object can withstand before slipping for a set of different object-hand relative poses. The experimental results show that our system is able to generate reliable predictions which agree with tests performed by manually pushing the object in the hand of the robot for previously unencountered grasp configurations.

Future directions of work include expanding our sensor modalities from kinesthetic perception to cover a wider spectrum of haptic perception (see Fig. 2) by use of tactile sensing. We also aim to incorporate into our system the estimation of the axis of rotation of the object in the hand of the robot as it can improve the results shown here. We have assumed a constant axis of rotation around the fingertips of the hand that might not correspond precisely with the actual axis around which the object rotates when it is manipulated. In order to cope with this issue, we aim to use adaptive control techniques previously used for estimating the kinematic constraints of hinged doors [19] and treat the object as a virtual hinge. We are also interested in coupling this work with probabilistic grasp assessment techniques and object categorization as demonstrated in our previous work in [20], [21].

Acknowledgment

This work has been supported by the European Union FP7 project RoboHow.Cog (FP7-ICT-288533), the Swedish Research Council (VR) and Swedish Foundation for Strategic Research (SSF). The authors gratefully acknowledge the support.

References

[1] R. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing - from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

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

[3] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, 2000, pp. 348 –353 vol.1.

[4] R. Howe and M. Cutkosky, "Sensing skin acceleration for slip and texture perception," in *Proc. IEEE International Conference on Robotics and Automation*, 1989, pp. 145–150.

[5] C. Melchiorri, "Slip detection and control using tactile and force sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 5, no. 3, pp. 235 –243, sep 2000.

[6] B. Heyneman and M. Cutkosky, "Biologically inspired tactile classification of object-hand and object-world interactions," in *Proc. IEEE International Conference on Robotics and Biomimetics*, 2012.

[7] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater, "Learning object-specific grasp affordance densities," in *IEEE 8th International Conference on Development and Learning*, 2009, pp. 1–7.

[8] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept. 2011, pp. 1554–1560.

[9] J. Gibson, *The theory of affordances*. In: Shaw R, Bransford J, editors. Perceiving, acting and knowing: towards an ecological psychology. Hillsdale, NJ: Erlbaum., 1977.

[10] A. Bicchi, J. Kenneth Salisbury, and D. Brock, "Experimental evaluation of friction characteristics with an articulated robotic hand," *Experimental Robotics II*, pp. 153–167, 1993.

[11] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, vol. 4, oct. 2003, pp. 3692 – 3697 vol.3.

[12] ——, "Grasp planning: how to choose a suitable task wrench space," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 1, april-1 may 2004, pp. 319 – 325 Vol.1.

[13] T. Sugaiwa, G. Fujii, H. Iwata, and S. Sugano, "A methodology for setting grasping force for picking up an object with unknown weight, friction, and stiffness," in *IEEE-RAS International Conference on Humanoid Robots*, dec. 2010, pp. 288 –293.

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

[15] J. B. Van Erp, K.-U. Kyung, S. Kassner, J. Carter, S. Brewster, G. Weber, and I. Andrew, "Setting the standards for haptic and tactile interactions: ISO's work," in *Haptics: Generating and Perceiving Tangible Sensations*. Springer, 2010, pp. 353–358.

[16] R. Howe, I. Kao, and M. Cutkosky, "The sliding of robot fingers under combined torsion and shear loading," in *Proc. IEEE International Conference on Robotics and Automation*, apr 1988, pp. 103 –105 vol.1.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[18] D. Song, C. H. Ek, K. Hübner, and D. Kragic, "Multivariate discretization for Bayesian Network structure learning in robot grasping," in *IEEE ICRA*, 2011, pp. 1944–1950.

[19] Y. Karayiannidis, C. Smith, F. Viña, P. Ögren, and D. Kragic, "Model-free robot manipulation of doors and drawers by means of fixed-grasps." in *IEEE International Conference on Robotics and Automation*, 2013.

[20] M. Madry, D. Song, and D. Kragic, "From Object Categories to Grasp Transfer Using Probabilistic Reasoning," in *IEEE International Conference on Robotics and Automation*, 2012.

[21] Y. Bekiroglu, D. Song, L. Wng, and D. Kragic, "A Probabilistic Framework for Task-Oriented Grasp Stability Assessment," in *IEEE International Conference on Robotics and Automation*, 2013.

A Probabilistic Framework for Task-Oriented Grasp Stability Assessment

Yasemin Bekiroglu, Dan Song, Lu Wang and Danica Kragic

***Abstract*— We present a probabilistic framework for grasp modeling and stability assessment. The framework facilitates assessment of grasp success in a goal-oriented way, taking into account both geometric constraints for task affordances and stability requirements specific for a task. We integrate high-level task information introduced by a teacher in a supervised setting with low-level stability requirements acquired through a robot's self-exploration. The conditional relations between tasks and multiple sensory streams (vision, proprioception and tactile) are modeled using Bayesian networks. The generative modeling approach both allows prediction of grasp success, and provides insights into dependencies between variables and features relevant for object grasping.**

I. Introduction

A lot of current work in robotics is inspired by human goal-directed behavior [1]. In humans, goal-directedness is obtained through multiple development stages, both through the sensorimotor *exploration* (trial and error) and through the *observation* of others interacting with the world (imitation learning) [2]. The former is addressing the problem of learning through self-experience in order to associate the sensorimotor signals to the direct motor effects. The latter involves human supervision, which is especially beneficial for efficient learning of complex tasks. Robotic approaches often focus on just one of these two aspects. Linking between the two is often through manual encoding [3] or applied to simple tasks [4], [5], [2], [6]. The main challenges originate from the differences in commonly adopted representations [7].

The gap between the representations is especially visible when dealing with robot grasping tasks. For example, if a robot is given a high-level task command, e.g., *pour me a cup of coffee*, it needs to make decision on which object to use, how the hand should be placed around the object, and how much gripping force should be applied so that the subsequent manipulation is stable. Several sensory streams (vision, proprioception and tactile) are relevant for manipulation. The problem domain and hence the state space becomes high-dimensional involving both continuous and discrete variables with complex relations. Traditional dynamic systems approaches in robotics e.g., [8] focus mainly on optimal planning and control of hand trajectories, hence the state space only includes kinematic parameters of the arm and the hand. The relations between many grasping-relevant variables mentioned above can not be addressed simultaneously.

Probabilistic frameworks based on graphical models have proved to be powerful in various fields with high-dimensional complex problem domains [4], [9], [10], [6]. Graphical models encode the relations between variables through their probabilistic conditional distributions. Such distributions do not require the variables to have the same underlying representations. Therefore, high-level symbolic variables such as task goals can be naturally linked to the low-level sensorimotor variables such as hand configuration. Furthermore, the model can be combined with the probabilistic decision making where grasp plan and control can be performed through inference even with noisy and partial observations [11].

Some recent work in the area [12] exploited these strengths and linked the grasp plan to the manipulation tasks through Bayesian networks (BNs). The work emphasized the geometric constraint of a task for planning grasps based on simulated vision inputs. Tasks, however, also require various manipulations: *pouring* needs rotating a bottle that contains liquid, and *hand-over* needs only parallel transportation. The stability demand therefore differs due to different manipulations requested by tasks.

In this paper, we integrate this task-dependency with stability assessment. A method combining self-exploration and supervision is implemented, where self-exploration enables the robot to learn about its own sensorimotor ability (how to grasp an object to stably lift and manipulate it), while human tutoring helps the robot to associate its sensorimotor ability to high-level goals. In particular, we use a probabilistic model to integrate the semantically expressed goal of a task with a set of continuous features. We present an extensive evaluation of the proposed approach on a real robot platform equipped with multiple sensory modalities (vision, proprioception and tactile). The results show that the proposed model accurately estimates grasp success both at the stage of planning (before execution in real environments) and during grasp execution.

II. Related Work

Planning and executing a grasp that is robust and stable is an important topic in grasp research (see [13] for a recent review). The quality measures of stability are mostly based on *force-closure* of a grasp wrench space. A force-closure grasp means that any disturbing external forces can be balanced by the forces applied at the contacts. However these approaches assume perfect knowledge of the contacts between the hand

Y. Bekiroglu, D. Song, L. Wang and D. Kragic are with the Centre for Autonomous Systems and the Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {yaseminb,dsong,luwang,dani}@kth.se. This work was supported by the EU through the projects eSMCs (FP7-IST-270212) and RoboHow.Cog (FP7-ICT-288533).

and the object, which is usually an unrealistic demand on real setups. On the other hand, experience based approaches where the robot learns good grasping configurations through real execution [14], [15], [16] have proved to be successful.

But a good grasp should not only be stable, it also needs to be suitable for the task, i.e., *what do you want to do after you lift the object*. Very few work has put effort on planning grasps in a goal-directed manner. Xue et al. [3] manually encoded the expertise about task semantics provided by a human tutor. A recent work [12] used Bayesian networks to learn the grasping task constraints that depends on a set of geometric attributes from both objects and grasps (e.g., hand positions). However manipulation tasks do not just concern geometric constraints. A *pouring* task not only requires the bottle opening to be unblocked, but also needs the grasp to be stable enough to rotate the bottle. We need to link task information with stability in real world scenarios.

A natural extension is to combine supervised task learning with experience-based stability learning. This allows stability to be assessed in a task-oriented manner. This is especially beneficial for energy-efficient control: when a task (e.g., *hand-over*) does not require strong grasping for difficult manipulations (e.g., waving for the *hammering* task), a relatively smaller gripping force can be applied. Combining task with stability was rarely studied. Some work [17], [18] defined task-related grasp quality measures which combined task knowledge with analytical stability measures used in traditional grasp stability studies. Such approaches therefore also suffer from partial and uncertain knowledge of the world in real setups.

Probabilistic learning is a powerful paradigm for modeling and reasoning about the noisy and uncertain real world data [4], [9], [10], [6]. For robot grasping, planning and control rely heavily on vision sensing with typically noisy and incomplete observations. Probabilistic approaches combining vision and tactile sensing [19] provided an on-line estimate of belief states which were used to plan the next action. Toussaint et al. [4] proposed a coherent control, trajectory optimization, and action planning architecture by applying the inference-based methods across all levels of representation. Montesano et al. [6] used Bayesian networks to learn object affordances, and applied them to goal-directed motion planning.

However, to our knowledge, no one has proposed a model that addresses both task-oriented grasp planning and stability-oriented grasp execution in real environments. In this paper we close the learn-plan-execute loop where the robot learns task knowledge from human teaching, and grounds this knowledge in low-level sensorimotor systems through self-exploration (manipulating the object) in a real environment. We use Bayesian networks to model conditional relations between task and stability knowledge with a multitude of features from vision (simulated in this work), proprioception, and tactile sensing. The generative modeling approach provides a flexible framework to guide detailed grasp planning and execution in a task-directed way.

III. MODELS

We use X to denote a set of features relevant for grasping tasks T. X originates from three groups of features, $\{O, A, H\}$, where O denotes an object feature set (from visual sensing), A denotes an action feature set that represents gripper configurations (from proprioception) and H denotes a haptic (or tactile) feature set. Detailed feature descriptions can be found in Section IV-B. We propose to use a generative approach, the Bayesian network [20], to model this grasp space. The goal is to apply the model for both task classification $P(T|X)$ and inferring the distribution of one variable conditioned on a task and other variables $P(X_i|T, X_j)$. $P(T|X)$ predicts how likely a grasp will succeed for a task, and $P(X_i|T, X_j)$ conveys domain knowledge such as the expected value of a tactile feature given a task and an object. To evaluate BN's classification performance, we compare it with a discriminative approach, Kernel Logistic Regression (KLR). In this section, we provide an overview of the two modeling approaches.

A. Kernel Logistic Regression

Kernel Logistic Regression is a nonlinear probabilistic classification model. Given a class variable (in this paper, the task T) and the input feature set (in this paper $X \subseteqq \{O, A, H\}$ as seen in Tab. I), KLR models the probability of the class variable $P(T|X)$ through a weighted sum of the similarities (kernels $\mathcal{K}$) between a testing point $\mathbf{x}$ and each training point $\mathbf{x}_i$ [21]:

$$p(\mathbf{t}|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp\{-\sum_{i=1}^{n} w_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i)\}} \tag{1}$$

In this paper we choose $\mathcal{K}$ to be a Gaussian kernel. Training a KLR model is to find the weight vector $\mathbf{w}$ that maximizes the regularized probability of the data

$$-\sum_{i=1}^{n} \log p(y_i|x_i; w_i) + \eta \,\mathrm{trace}(\mathbf{w}\mathbf{K}\mathbf{w}^T) \tag{2}$$

where $\mathbf{K}$ is the kernel Gram matrix, with $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, and η is the regularization constant. During training, the kernel bandwidth parameters and η are chosen by cross-validation.

B. Bayesian Network

A Bayesian network [20] is a probabilistic graphical model that encodes the joint distribution of a set of random variables $V = \{V_1, V_2, \ldots, V_m\}$. Each node in the network represents one variable, and the directed arcs represent conditional independencies. Given a structure of the network S and a set of local conditional probability distributions (CPDs) of each variable V_i, the joint distribution of all the variables can be decomposed as

$$p(\mathbf{v}) = p(\mathbf{v}|\boldsymbol{\theta}, S) = \prod_{i=1}^{m} p(\mathbf{v}_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S) \;, \tag{3}$$

where $\mathbf{pa}_i$ denotes the parents of node V_i, and the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_m)$ specifies the CPDs. Learning a

BN includes discovering from a dataset: *1)* how one variable depends on others ($\boldsymbol{\theta}$), and *2)* what the conditional in-dependencies between different variables are (S). The former is an instance of parameter learning and the latter of structure learning. Various algorithms and techniques have been developed to learn a BN in different model and data conditions (see [22] for a review).

In this paper, we use the Bayesian network to model the joint distribution of a set of task and stability-relevant variables (see Tab. I), i.e., $V = \{T, X\}$ where $X \subseteq \{O, A, H\}$. To correctly describe a grasping task, both conceptual high-level information and continuous low-level sensorimotor variables are required. The variables in this work are both discrete (e.g., $task, obcl$), and continuous (most O, A, H features). The continuous features such as hand grasp configuration can be high-dimensional with complex probabilistic distributions.

Learning BN structures from both continuous and discrete data is an open problem, particularly when continuous data is high-dimensional and sampled from complex distributions. Most algorithms for structure learning only work with discrete variables. Therefore, a common approach is to convert the mixed modeling scenario into a completely discrete one by discretizing the continuous variables [23]. In this paper we use a two-step discretization scheme. For a high-dimensional continuous variable X, the data in original observation space is first projected to a low-dimensional space, and then a parametric mixture model (multi-variate Gaussian mixture) is learned to model the data density in this space,

$$p(\mathbf{x}) \propto \sum_{k=1}^{M} \lambda_k N(\mathbf{x}|\mathbf{u}_k, \Sigma_k^{-1}). \tag{4}$$

where $\mathbf{u}_k$ and Σ_k are the mean and covariance of each Gaussian component, and λ_k is the mixing proportion. The parameters of the mixture model are learned using the standard EM approach. The number of the clusters for each variable is found through cross-validation where the task classification performance with the BN is maximized.

We use a greedy search algorithm to find the network structure (the directed acyclic graph, or DAG) in a neighborhood of graphs that maximizes the network score (Bayesian information criterion [24]). The search is local and in the space of DAGs, so the effectiveness of the algorithm relies on the initial DAG. As suggested by Leray and Francois [25], we use another simpler algorithm, the maximum weight spanning tree [26], to find an oriented tree structure as the initial DAG.

C. Inference in Bayesian Networks

A trained network defines the factorization of the joint distribution of the observations, $p(V) = p(T, O, A, H)$, in terms of a graph of conditional dependencies. We can now compute the posterior distribution of one or group of variables given the observation of others. A common way for doing this is to apply the junction tree algorithm [27]: an algorithm of local message passing to compute the distribution of the variables of interest. The output of the network is a multinomial distribution over each of the discrete states of the network,

$$p(\mathbf{v}_i \rightarrow \mathbf{u}_{ik}|\Pi_i = \mathbf{U}_j). \tag{5}$$

stating as "*the probability of variable V_i is at its discrete state $\mathbf{u}_{ik}$ when a set of other variables Π_i is observed to be at the state $\mathbf{U}_j$*".

D. Generative Model

A Bayesian network is a generative model where not only the class probabilities $p(T|X)$ can be inferred as KLR, but also the class conditional distributions can be predicted $p(X|T)$. The former means we can use a BN to predict success of a grasp to achieve a task given observed object and action features by inferring the posterior distribution $p(T|O, A)$, i.e., to classify T. The latter means that we can also find, given an assigned task, the posterior distribution of the object $p(O|T)$ and/or grasp features $p(A|T, O)$. This provides the basis for the robot to select objects that afford a given task, e.g., *something to drink from*, and plan an optimal grasp strategy using the object to fulfill the task requirements.

In addition, Bayesian networks allow us to infer the domain knowledge through data. The network structure depicts an influence diagram illustrating the conditional relations between different variables. Also the class conditional on feature variables provides an intuitive evaluation of task and stability-related requirements.

Another strength of the BN is its ability to infer the grasp success with partial observation. In a task-based grasp adaptation scenario (see Fig. 7), this is especially important because we can predict the grasp success given observation on only object features and grasp parameters planned in a simulation environment. Grasp replanning therefore can be initiated without having to execute an unstable grasp using real robot platforms. Though this can also be done using discriminative models, each observation condition requires training of a separate model.

IV. Modeling Sensor Data and Data Acquisition

We will first describe the data acquisition process which uses both a grasp simulation environment and a real robot platform. We then present a detailed description of the sensory data representation.

A. Data Acquisition

The goal of the data acquisition is to obtain a set of data that instantiate the variables in $\{O, A, H, T\}$. We use a 7-dof Schunk dextrous hand equipped with tactile array sensors. The hand is attached to a 6-dof Kuka arm that is mounted on a robust shelf. Seven home-environment objects including three bottles and four mugs are used for the data generation. In GraspIt! [28] a Schunk hand model is used for planning grasps on the corresponding object models and extracting features. The seven object models that capture similar sizes and shapes of the real objects can be seen in Fig. 3.

Fig. 1 shows the schematic of the data generation process. To extract the features in Tab. I, we first generate grasp hypotheses using the grasp-planner BADGr [29]. Each grasp

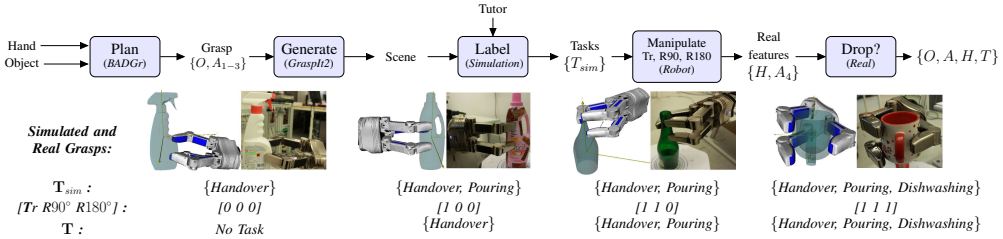


Fig. 1. Data Generation Process: The top row is a diagram of the process. The bottom row shows four example grasps and how they are labeled with different tasks. The three tasks are hand-over, pouring and dishwashing, each of which has to satisfy one of the stable manipulations: transport (Tr), 90° rotation (R90), and 180° rotation (R180), respectively.

hypothesis is first visualized in GraspIt! by a human tutor who associates it with a task label from the simulation (T_{sim}). Then the hypothesis that is good for at least one task T_{sim} is used on the robot platform to perform a set of grasps and manipulations on the similar real object during exploration (see Fig. 2). If a grasp that is considered to be good for a task, e.g., *pouring* (by label T_{sim}) results in an unstable 90°rotation (object drops/slips) which is defined to be the required manipulation for *pouring* task, then it will be considered to be bad for *pouring* in the final task label T. In this paper, we experiment with three tasks: *hand-over*, *pouring*, and *dishwashing*. These tasks are associated with certain manipulations and geometric constraints: For *hand-over*, the object should be transported (Tr) horizontally and the grasp should leave enough uncovered surface or handle for safe regrasp. For *pouring* the object should be rotated 90° (R90) and the grasp should not cover opening part for pouring the liquid. For *dishwashing*, the object should be rotated 180° (R180) and the grasp should allow placing the object upside-down.

During data generation, our goal is to execute the planned hypotheses around the object (see Fig. 3). To avoid that some grasps are not reachable, we place the object in a known location in front of the robot, and manually rotate the object along the vertical axis by a 45° increment to place the hypotheses in the robot's working space.

Because of the uncertainty introduced in both the motor system and the manual placement, the real hand pose will not precisely represent the values generated in simulation. This uncertainty is simulated by adding zero-mean and Gaussian-distributed noise to the variables. Fig. 3 shows example grasps generated on bottles and mugs in both clean (left) and noisy (right) versions. The resulting grasping position has noise with standard deviation about 0.4 to 1.1 (cm) in the three dimensions. For each task a grasp dataset with equal number of positive and negative samples is obtained. The number of positive samples is 1026 for *hand-over*, 1143 for *pouring* and 831 for *dishwashing*. These three datasets are used to train task-specific BNs.

B. Feature Description

Tab. I lists all the features used for the representation of the sensor data. The object features O include the object class identity $obcl$, the three dimensional $size$, and the convexity $cvex$. The action features A describe the hand pose (position and orientation) in the object-centered coordinate system and the final hand configuration $fcon$. We decompose the grasp position into a unit sphere $npos$ and the radius rad for visualization purpose in the inference results.

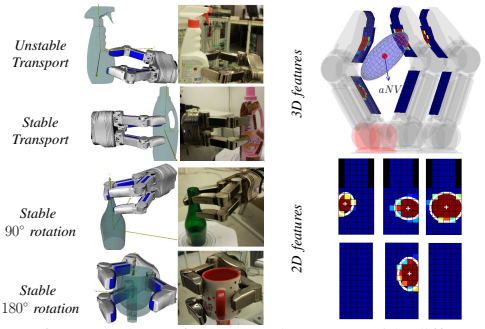


Fig. 2. Left panel shows four example grasps with different stability conditions: Simulated grasps are executed on the real platform and for each grasp three manipulations are applied: Horizontal transport, 90° rotation and 180° rotation. Each real grasp is marked with the stability outcome of the manipulation. Right panel illustrates tactile related features.

In terms of haptic features H, we calculate a set of tactile features (see Figure 2) measured by the six tactile sensors on the Schunk hand. iG carries information about the distribution of the pressure in the vertical and the horizontal directions and also the pressure centroids (iC) locally for each sensor array. It is calculated based on image moments up to order 2. pG is the 3D version of iG with respect to the wrist frame, it is calculated using $fcon$ and it represents the pressure distribution and the pressure centroid (pC) considering all the sensors. Another tactile feature is the average normal vector aNV that is calculated by $\sum_{i=1}^{486} \tau_i r_i$ where r_i is the normal vector of the texel i and τ_i is the normalized tactile reading in the texel i ($\sum_{i=1}^{486} \tau_i = 1$).

We emphasize that the representation of a grasp may be redundant, e.g., iG contains information of e.g., iC. Such an over-representation of the feature variables allows us to select the most representative variables and enables efficient learning and inference. It also allows us to use BNs to identify the importance of, and the dependencies between these variables in various scenarios of robot grasping tasks.

V. Model Selection

Model selection is a process including three steps: 1) dimension reduction, 2) variable selection (using the low-dimensional representation), and 3) optimizing data discretization. These three steps are applied on the three datasets

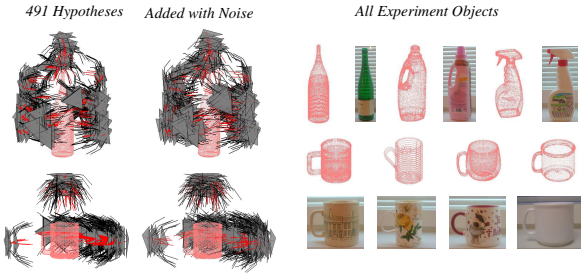


Fig. 3. Data Collection: The left panel shows grasp examples generated on the two classes of objects (mugs and bottles). The right panel shows all the objects.

TABLE I

FEATURE SET WITH DIMENSIONALITY D (LOW/HIGH) AND THE NUMBER OF DISCRETE STATES M (OPTIMIZED FOR EACH OF THE THREE TASKS [HAND-OVER, POURING, DISHWASHING] AND SHOWN FOR THE SELECTED FEATURES). T, O, $A_{1,2,3}$ ARE FROM THE SIMULATION, A_4 AND H ARE FROM THE REAL ROBOT.

	Name	D	M	Description
T	$task$		2	Binary task identifier
O_1	$obcl$		2	Object class
O_2	$size$	3		Object dimensions
O_3	$cvex$	1	[5, 5, –]	Convexity value $[0, 1]$
A_1	dir	4		Quaternion hand orientation
A_2	$npos$	3	16	Unit grasping position
A_3	rad	1	[15, 14, 14]	Radius of $npos$
A_4	$fcon$	2/7	[7, 7, 7]	Final hand configuration
H_1	iG	5/30		2D pressure distribution
H_2	iC	3/12	[11, –, –]	2D pressure centroid
H_3	pG	3/9		3D pressure distribution
H_4	pC	3		3D pressure centroid
H_5	aNV	2/3	[–, 5, 5]	Average normal vector

separately for each task and task-specific BNs with binary task variables are built.

1) Dimensionality Reduction: There are many techniques for dimension reduction [30]. Ideally a cross-validation process should be used to select optimal technique and their parameters. However, we have many steps for model selection, a full-scale model selection will be expensive. Considering the main focus of the paper is not to evaluate dimension reduction techniques, we decide to select a single method. We choose Kernel PCA [31] because of its capability to model non-linear manifolds which is a character of our problem domain. Tab. I shows the resulting dimensionality together with the original dimensionality on a set of variables.

2) Variable Selection: We use the HITON algorithm [32] to perform the optimal variable selection for the three tasks. HITON works by first inducing the Markov Blanket of the target variable to be classified. In this paper the target is the binary task variable T, and its Markov Blanket is denoted by $MB(T)$. Then support vector machine is used to further remove the unnecessary variables in the $MB(T)$ in a greedy hill-climbing fashion. The performance metric is the task classification rate. Exhaustive search through all subsets of features returned in $MB(T)$ is prohibitive, so we adopt a set of heuristics to form a smaller search space: 1) the subset must include $obcl$ and $npos$ because we are interested in inferring the conditionals involving these variables, 2) there must be at most two features in each of the O, A and H feature sets. We adopt a stopping point at a 95% threshold of classification accuracy. The subset of features with the highest score discovered up to this point is selected as the satisfactory set of features. Fig. 4 shows which variables have been selected for each of the three tasks.

3) Optimizing Data Discretization: This is a step for only Bayesian networks. The structure learning requires discrete data. However, this leads to loss of information. When the resolution is low (i.e., a few discrete states), the variance in the original continuous domain that is discriminative may be smoothed out. On the other hand, for the variables that are not discriminative, a high resolution will jeopardize the classification performance due to the curse of dimensionality. We therefore want to find an optimal granularity M in Eq. (4)), on a small set of variables ($\{cvex, rad, fcon, iC, aNV\}$). The optimal granularity maximizes the task classification performance with the BNs. Tab. I shows the resulting number of discrete states M for each of the three tasks.

VI. MODEL EVALUATION

We evaluate the Bayesian network-based modeling framework in two aspects: classification performance, and how we can use the generative model for understanding the problem domain.

For classification performance, we compare the BN modeling with the discriminative approach KLR under two observation conditions: the partial observation when only simulated object and action variables are observed ($T|O, A_{1,2,3}$), and the full observation when haptic information and A_4 are also available after grasp execution in the real environment ($T|O, A, H$). We perform this over 50 trials with 20% hold-out splits.

Under these conditions, 50 trials of cross-validation with 20% hold-out splits are performed. In each trial, for each task three models are trained: 1) $KLR(O, A, H)$ with all the selected variables, 2) $KLR(O, A_{1,2,3})$ with only simulated variables, and 3) $BN(O, A, H)$ with all the selected variables. We do not need to train BN with only simulated variables ($BN(O, A_{1,2,3})$) because the task probability can be inferred in BNs with partial observations. When training KLR models, we use the continuous low-dimensional representation. And when training BNs, we use the optimal discrete data. In each trial, both structure and parameters of the BNs are learned. Since each trial uses different set of training data, the resulting structure can be different.

For each task, the inference results on two variables are shown: $npos$ and one of the selected H features for the task. We chose one tactile-related feature to show that the BN can be used to produce an expectation over sensor data given task constraints. For each variable, we evenly sample a set of points $\mathbf{x}$ in the low-dimensional space for easy visualization. For each sampled point, a conditional likelihood is obtained given the three tasks and the object class $p(\mathbf{x}|task, obcl)$ to generate the *likelihood maps* seen in Figure 6.

A. Network Structures

Fig. 4 shows the Bayesian network structures (DAGs) with the highest task classification performance for the three tasks.

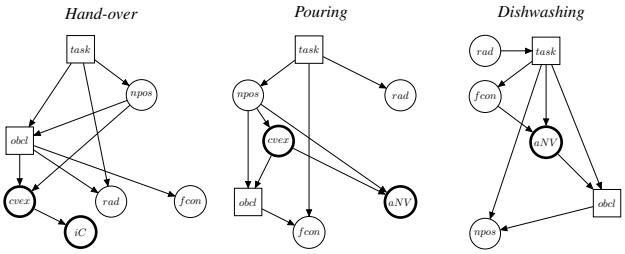


Fig. 4. The structure of BNs with the highest classification performances: DAGs of the three BNs each of which models one binary classification of one task. Square nodes represent discrete variables and circled nodes continuous ones. The differences in variable selection among the three tasks are highlighted by thick border of the nodes.

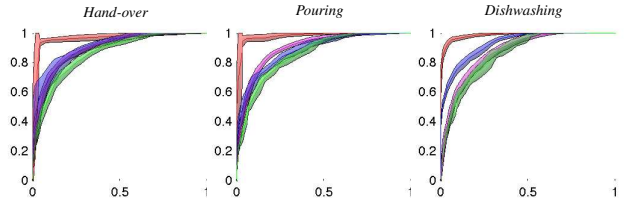


Fig. 5. Classification: The average ROC curves for three tasks. Red is KLR with full observation (O, A, H). Pink is KLR with partial observation (O, A_{1-3}). Blue is BN with full observation (O, A, H). Green is BN with partial observation (O, A_{1-3}). The transparent regions represent the one standard deviation of the true positive rate.

The represented nodes in each network are the variables selected using the HITON algorithm [32]. The differences in selected variables between different tasks are highlighted by the thick-bordered nodes.

Considering haptic features H, *hand-over* task selects iC, whereas *pouring* and *dishwashing* tasks both select aNV. iC is a feature characterizing the local pressure centroid of each tactile sensor pad on the fingers, whereas aNV summarizes the overall pressure distribution considering all the sensors and also the finger configurations. In other words, aNV encompasses stronger information that may be relevant to stability especially when the task demands stronger grasping such as *pouring* or *dishwashing*.

As to the network structure, all the three tasks have direct conditional relations with $npos$ and rad. This is natural since the position of the hand relative to the object is an important factor influencing both the affordance of a task (from which direction to approach the object $npos$), and its stability requirements (how far away the hand is from the object center of mass rad). For *dishwashing* T is directly connected to aNV, whereas for *pouring* T influence aNV through $npos$. This may be due to that *dishwashing* requires a manipulation with $180°$ rotation, which, compared to $90°$ rotation for *pouring*, is more demanding in terms of grasp stability. So the task success for *dishwashing* depends on aNV even if the $npos$ is also observed.

B. Classification

The area (AUC) under the ROC curve is used as the performance metric. The ROCs are derived by thresholding the classifier outputs, the probability of task success $p(T = \text{true}|X)$. Figure 5 shows the ROC curves for task classification results averaged over 50 trials. Table II shows the mean and the standard deviation of the AUCs.

In general, the BNs with both full and partial observations have good classification performances for all the three tasks. Under full observation, KLR models perform better than BNs. However, we note that when the real sensor data (H and A_4) are not observed, KLR's performance drops a lot compared to BNs. To confirm this, we conduct a two-sample t-test on the AUC scores over 50 trials of the experiment. The hypothesis is: "*The classification performance with full observation is* 0.07 *higher than the performance with partial observation*", briefed as "full $\triangle 0.07 >$ partial". The results show that at the significance level 0.05, the hypothesis is accepted for the KLR, but rejected for the BN. In other words, KLR with partial observation performs similarly to BN with both observation conditions. Another result is that, when real sensory features H and A_4 are not observed, the performance drop for *dishwashing* task in the BN is higher than for the other two tasks. This is related to the differences in the task requirements of grasp stability which has explained the structural differences depicted in Fig. 4. For example, when aNV is not observed in *dishwashing*, $p(T|X)$, more useful information is lost than in *pouring*. Overall, BN modeling provides high classification results. We prefer BNs since they allow inference on any variable given full or partial observation of others. KLR requires training separate models for different observation conditions.

C. Inference

Fig. 6 shows likelihood maps in relation to different features, tasks and object categories. The brighter color indicates higher probability of a successful grasp. On the left side, we can see the results on $p(npos|task, obcl)$, where the hand positions in the object frame are projected on the unit sphere. For the *pouring* task, the robot should not grasp the mugs or the bottles from the top, which is reflected by the dark color on the $npos$ sphere. However, top grasps are allowed for *hand-over* task. Among the two object classes, only the mugs afford *dishwashing* task, which is indicated by the fact that the likelihood maps are almost completely black.

On the right side, the results of the two tactile features projected on the low dimensional space, 3D $p(iC|task, obcl)$ for *hand-over* task and 2D $p(aNV|task, obcl)$ for the other two tasks are seen. We observe clear differences in these "haptic images" both between the two different object classes, and also between the different tasks. This reflects different "haptic expectations" given task and object conditions. For the pouring task, we observe that the mugs has a clear cut between "bad" and "good" regions in the aNV map, whereas the bottles have more gradual change in the likelihood map. The reason may be that the bottles are much taller than the mugs therefore there are more grasps along the longitudinal direction on the bottles that have gradual changes in grasp quality.

TABLE II

MEAN AND STANDARD DEVIATION OF AUCS FOR THE THREE TASKS.

Task	KLR full	KLR partial	BN full	BN partial
Hand-over	0.97 (0.01)	0.90 (0.01)	0.90 (0.04)	0.86 (0.01)
	$\triangle 0.07 > *$			
Pouring	0.98(0.01)	0.90 (0.01)	0.88 (0.02)	0.86 (0.02)
	$\triangle 0.07 > *$			
Dishwashing	0.98 (0.01)	0.87(0.02)	0.92 (0.01)	0.86 (0.02)
	$\triangle 0.07 > *$			

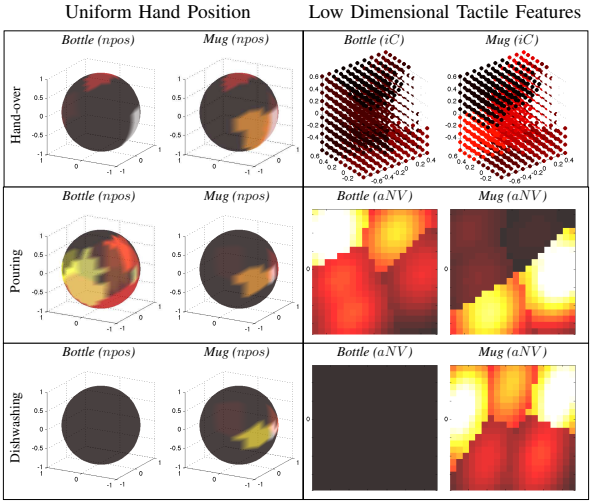


Fig. 6. Inference: The likelihood maps of the continuous variables conditioned on task and object class. Left side in inference results shows $p(npos|task, obcl)$ for all the three tasks. On the right side, $p(iC|task, obcl)$ is obtained for *hand-over* and $p(aNV|task, obcl)$ is obtained for the other two tasks.

D. Model Application

We conclude the paper by a task-oriented, stability-based grasp adaptation scenario. The goal is to demonstrate one way of applying the proposed probabilistic framework. Fig. 7 depicts a two-step grasp adaptation process, where the first step predicts if a planned grasp hypothesis affords an assigned task (from the simulated $O, A_{1,2,3}$ features) before it is executed on the real robot, and the second step predicts if the grasp affords manipulation demanded by the task once the grasp has been executed. Here the sensory inputs H and A_4 are available which allows more accurate prediction with the full observation $p(T|O, A, H)$ before the object is lifted. Such a *double-guarded* system is beneficial to efficiently plan and execute the robot grasping.

Fig. 8 demonstrates a grasp adaptation process for the input *pour with this detergent bottle*. The top row shows the grasp hypotheses sequentially produced by a planner. Before they are executed on the real robot platform $p(T|O, A_{1,2,3})$ rejected the first three hypotheses. This is reflected by the location of data point (green dot) in the dark region of $npos$ likelihood maps. The grasp replan is triggered until the fourth hypothesis is found to be good for grasp execution. It is however predicted to fail under the full observation $p(T|O, A, H)$ (aNV is in the dark region of the likelihood map). A replan is again triggered until a good grasp is found with the full observation.

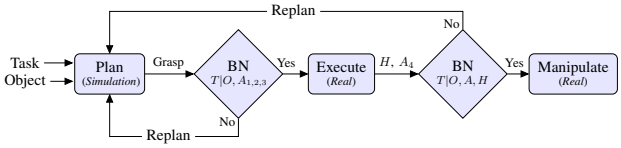


Fig. 7. Application Diagram: Task-based grasp adaptation.

VII. CONCLUSION

We have proposed a unified probabilistic framework using Bayesian networks to assess grasp stability in a task-oriented manner. The framework enables combination of human supervision and self-exploration during manipulation to encode task-dependent stability requirements. The learned network successfully predicts outcomes of a grasping action both in terms of the geometric requirements and in terms of the stability demands for the subsequent manipulations. Since the high-level task goals are seamlessly linked to low-level haptic sensory outputs, grasp planning and control are efficiently entwined. In addition, the generative model allows us not only to predict grasp success and task relevance, but also convey domain knowledge. We can infer structural dependencies between different variables, and form conditional expectations on various sensory features. In other words, we can reason on which sensory features are most relevant for a specific task and the robot can perform on-line decision making on what-to-measure, thus optimizing the use of sensory data.

The work opens an interesting avenue for future research.

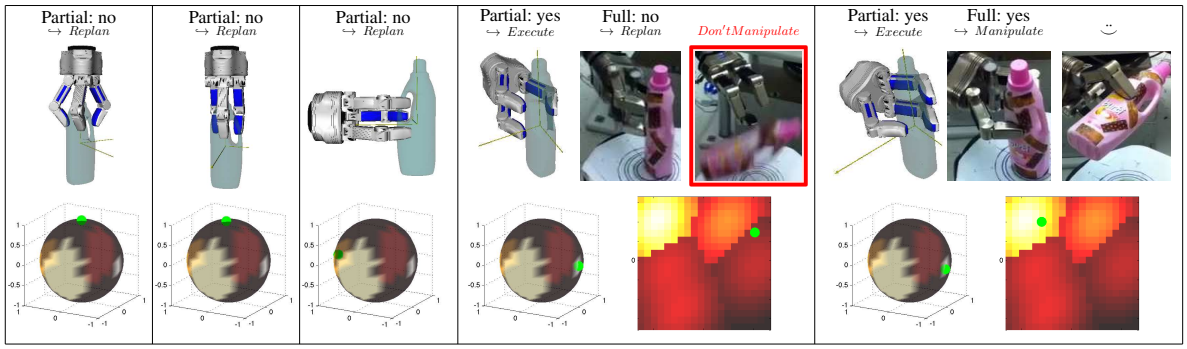


Fig. 8. Application: Two-loop grasp adaptation when task is *pouring* with the detergent bottle following the flowchart in Fig. 7.

The current system relies on a single time instance for decision making and this is at the point of grasp completion. Although this may be sufficient for many tasks, temporal data is more informative for cases where on-line grasp adaptation and control is needed. For example, if the shape of the object is very complex or if the mass distribution is changing rapidly. We can explore dynamic models such as Dynamic Bayesian Networks for this purpose. In our approach, robot grasping is not a stationary process and it may need further on-line adaptation. Task requirements may vary given different contexts or environments. In addition, sensory measurements may also change over time, requiring model update. One of the areas of future research is development of learning algorithms that allow incremental data discretization and structure update.

REFERENCES

[1] A. N. Meltzoff, *Elements of a Developmental Theory of Imitation*. Cambridge, MA, USA: Cambridge University Press, 2002, pp. 19–41.

[2] R. Rao, A. Shon, and A. Meltzoff, "A Bayesian Model of Imitation in Infants and Robots," in *Imitation and Social Learning in Robots, Humans, and Animals*, 2004, pp. 217–247.

[3] Z. Xue, J. Zoellner, and R. Dillmann, "Automatic Optimal Grasp Planning based on Found Contact Points," in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2008, pp. 1053–1058.

[4] M. Toussaint, N. Plath, T. Lang, and N. Jetchev, "Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[5] E. Oztop, D. Wolpert, and M. Kawato, "Mental State Inference using Visual Control Parameters," *Cognitive Brain Research*, vol. 22, no. 2, pp. 129–151, 2005.

[6] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[7] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, "ObjectAction Complexes: Grounded abstractions of sensorymotor processes," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 740–757, Oct. 2011.

[8] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1398–1403.

[9] B. Douillard, D. Fox, F. Ramos, and H. D. Whyte, "Classification and Semantic Mapping of Urban Environments," *The Int. Journal of Robotics Research*, vol. 30, no. 1, pp. 5–32, 2010.

[10] T. Hospedales and S. Vijayakumar, "Structure inference for Bayesian multisensory scene understanding." *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 12, pp. 2140–2157, 2008.

[11] M. Toussaint, L. Charlin, and P. Poupart, "Hierarchical POMDP Controller Optimization by Likelihood Maximization," in *Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2008.

[12] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning Task Constraints for Robot Grasping using Graphical Models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[13] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326 – 336, 2012.

[14] R. Detry, E. Başeski, M. Popović, Y. Touati, N. Krüger, O. Kroemer, J. Peters, and J. Piater, "Learning Continuous Grasp Affordances by Sensorimotor Exploration," in *From Motor Learning to Interaction Learning in Robots*. Springer-Verlag, 2010, pp. 451–465.

[15] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 1554–1560.

[16] L. Montesano and M. Lopes, "Learning Grasping Affordances from Local Visual Descriptors," in *IEEE Int. Conf. on Development and Learning*, 2009.

[17] J. Aleotti and S. Caselli, "Interactive teaching of task-oriented robot grasps," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 539–550, 2010.

[18] Z. Li and S. Sastry, "Task oriented optimal grasping by multifingered robot hands," in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, vol. 4, Jan. 2003, pp. 389–394.

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

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

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

[22] D. Heckerman, "A Tutorial on Learning With Bayesian Networks," Microsoft Research, Tech. Rep., 1996.

[23] L. D. Fu and I. Tsamardinos, "A Comparison of Bayesian Network Learning Algorithms from Continuous Data," in *AMIA*, 2005.

[24] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[25] P. Leray and O. Francois, "BNT Structure Learning Package: Documentation and Experiments," Université de Rouen, Tech. Rep., 2006.

[26] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[27] C. Huang and A. Darwiche, "Inference in Belief Networks: A Procedural Guide," *Int. Journal of Approximate Reasoning*, vol. 15, pp. 225–263, 1994.

[28] A. T. Miller and P. K. Allen, "GraspIt! A Versatile Simulator for Robotic Grasping," *IEEE Robotics and Automation Magazine*, 2004.

[29] K. Huebner, "BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems: Workshop on Grasp Planning and Task Learning by Imitation*, 2010.

[30] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality Reduction: A Comparative Review," 2008.

[31] L. J. P. van der Maaten, "Matlab Toolbox for Dimensionality Reduction (v0.7.1b)," 2010.

[32] C. F. Aliferis, I. Tsamardinos, and A. Statnikov, "HITON: a novel Markov Blanket algorithm for optimal variable selection." *AMIA Annu Symp Proc.*, pp. 21–25, 2003.