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

DR 2.3:
Representation of Gaps in Object Knowledge

Michael Zillich, Johann Prankl, Markus Vincze, Yasemin Bekiroglu, Danica Kragic, Sebastian Zurek, Rustam Stolkin

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

Due date of deliverable:	July 31 2010
Actual submission date:	July 30 2010
Lead partner:	TUW
Revision:	final
Dissemination level:	PU

Knowledge about objects plays an important role in many tasks to be performed by a cognitive agent. Like all knowledge acquired in interaction with the real world, object knowledge will contain uncertainties and will typically be only partially complete, at least initially. So object knowledge must be built up incrementally and continuously based on noisy measurements and guided by identified gaps in the models acquired so far. Moreover actions based on available object knowledge, such as grasping, must be able to cope with these uncertainties. However, object knowledge is also rather a broad and task dependent term and difficult to simply formulate in a unified form. In this report we present work on various aspects of object knowledge related to acquiring and extending visual object models, representing grasp stability under uncertain object knowledge and acquiring knowledge about object behaviour during simple manipulations.

Executive Summary

Object knowledge is a broad term and encompasses everything the system needs to know about objects in order to perform its tasks. This includes visual tasks (detection, recognition and tracking) and manipulation tasks (grasping, pushing). Depending on the task, different representations of object knowledge and associated gaps are suitable. Crucially all these representations must support representing uncertainty and incomplete knowledge to allow the system to self-understand (identify knowledge gaps) and self-extend (acquire missing information to complete models). This report presents accumulation of knowledge and handling of knowledge gaps within different aspects of object knowledge. Accordingly this deliverable is structured around these different aspects of object knowledge: acquiring and completing visual object models, identifying stable grasps under uncertain object knowledge, and modelling knowledge about physical object behaviour in simple manipulation tasks such as pushing.

Closely related to this work is the work on gaps in categorical knowledge reported in Deliverable DR.5.3. "Representations of gaps in categorical knowledge", which deals with identifying gaps regarding learned object properties, such as colour categories. Moreover related Deliverable DR.1.2 "Unifying representations of beliefs about beliefs and knowledge producing actions" presents the system-wide picture regarding knowledge, gaps and knowledge producing actions and contains additional details.

Role of object knowledge in CogX

Knowledge about objects in one way or another is central to almost any task a cognitive agent wants to perform. From identifying and picking up objects as part of fetch-and-carry tasks, establishing common ground in discourse with a human operator to the labelling of room types from typical objects found in a room. Common to all these scenarios are the facts that object knowledge is never complete and that available sensory information is never perfect. So the agent must be able to continuously update its knowledge, identify where models are incomplete and incorporate uncertain information in a probabilistic manner.

Contribution to the CogX scenarios and prototypes

As argued above object knowledge is part of all CogX scenarios. However, Dexter is the scenario that explicitly deals with objects, whereas in the other scenarios (Dora and George) objects play a less prominent role embedded in navigation and discourse tasks. Accordingly the work presented in this

report is related most clearly to the Dexter scenario, where the system learns the behaviour of objects under manipulation.

1 Tasks, objectives, results

1.1 Planned work

This deliverable reports work related to Task 2.10:

> **Task 2.10:** *Representations of gaps in object knowledge and manipulation skills. We will develop representations of the incompleteness of, and uncertainty about, models of objects. This is a prerequisite for reasoning about information-gathering actions and performing introspection. This task will feed into the unifying work on this in WP1. (M1 - 48)*

Task 2.10 is an ongoing task spanning the whole project and underlying all work on representation of object knowledge. I.e. all representations must be able to support identification of incompleteness and uncertainty. Different tasks will require different types of representations along with different types of knowledge gaps. This report summarises work on representations and knowledge gaps performed in the first two years.

1.2 Actual work performed

The following sections describe what types of object knowledge are used within the system, how uncertainties or knowledge gaps arise and are represented as well as identified and also partly acted upon.

1.2.1 From attention to proto-objects

Looking at a novel unknown scene certainly constitutes a major knowledge gap, basically "What is there?". So the first "gap" in object knowledge is to identify objects in the first place. In the absence of given object models the scene can not be segmented into meaningful entities. Bottom-up attention however provides a cue of where objects might be found so that models can be subsequently learned. So attention is a standard behaviour for narrowing knowledge gaps in the absence of any other information. While many traditional attention operators are based on 2D saliency measures and output regions that are likely to be attended by humans, we are interested specifically in likely object locations in the context of an exploring robot, where the 3D structure of the scene carries the most relevant information. To this end we use a 3D plane pop-out attentional operator [29] and [30] (DR.2.2 - Annex 2.4). This operator identifies supporting surfaces from 3D stereo points clouds as well as spaces of interest (SOIs) sticking out from these surfaces. Nothing is known yet about the content of these SOIs, other than the raw 3D data. So the initial big knowledge gap was broken down into several smaller knowledge gaps which can be subsequently processed.

Generated SOIs can now trigger either an attempt to recognise their content based on known shape primitives or already learned models using e.g. the detector or recogniser presented in [18] (DR.2.2 - Annex 2.2) or to refine these SOIs to support further interpretation. In the latter case we observe SOIs over time to first identify stable SOIs and then refine the coarse segmentation based on (typically noisy) 3D stereo data by performing graph cut segmentation [2] on the projected 2D image. We use colour as well as spatial cues from sampled foreground (sticking out) and background (supporting plane) points to precisely segment the object contour, which is then back-projected into the original 3D data. We refer to these refined SOIs as proto-objects, as the refined knowledge about precise contour (and thus interior) allows referring to object properties such as colour and shape. Learning and identifying gaps in such categorical knowledge are the subjects of Deliverables DR.5.2 "Continuous learning of cross modal concepts" and DR.5.3 "Representations of gaps in categorical knowledge" respectively.

1.2.2 Detection, recognition and tracking of objects

As indicated above, having selected spaces likely to contain an object we can now close the knowledge gap with respect to its content by identifying shape primitives (and instantiating new object models) or identifying already learned object instances.

In the former case we extract groups of edges representing 2D projections of a limited class of shape primitives and use the supporting ground plane to construct metric 3D wire frame object shape models [24, 23]. The saliencies of the gestalt principles underlying the grouping processes provide a confidence measure for the extracted shapes. These edge based wire frame models only capture the object geometry and lack visual appearance information, especially of the occluded backside which was constructed based on symmetry assumptions about the shape primitives. So we now learned a new object, but with a rather weak model. Detection of edge based models suffers heavily from background clutter and more detailed models are needed to allow recognition in tough real world scenes, with changing scale, shadows etc.

Recognition of objects is based on highly distinctive SIFT features and encompasses object identity as well as object pose. We use two approaches. One is based on a set of per-view bag-of-features object models to reason amongst a set of several hypotheses regarding object identities and views. Given a sub-image as provided by a projected SOI, the recogniser outputs a discrete probability distribution over objects (and their views), also taking into account the case of an unknown object. The recogniser thus represents uncertainty about object identity given an object image location.

The second approach recognises and locates object instances in 3D [18] (DR.2.2 - Annex 2.2). It is based on associating SIFT features with their lo-

cation on the 3D object surface as provided by the above wire frame models. Online learning of these object models is covered in the following section. A RANSAC based robust pose estimation scheme then outputs a 6D object pose (independently for each known object) as well as a likelihood measure of object identity based on the ratio of matched SIFT descriptors.

Assuming known identity of the object and thus a known 3D model, the remaining uncertainty relates to its 6D pose, which we represent by a parameter free probability distribution function. Model based 3D tracking of objects is based on a particle filter [19], which approximates the estimated PDF using a set of particles. The tracker uses the above "sparse" wire frame models and extends them with edges extracted from surface texture to increase robustness to clutter and occlusion. The shape of the 6D pose PDF is unconstrained, allowing the representation of "small" uncertainties related to discretisation noise or image blur as well as distinctive alternative solutions (think of a flipping Necker cube) via multi-modal PDFs.

Together the above processes fill in more and more object knowledge, from rough location based on attention, over contours and wire frame models, to detailed appearance models with associated pose, thus gradually reducing knowledge gaps. Remaining uncertainties are represented as continuous or discrete probability distributions or in simpler cases as confidence measures.

1.2.3 Acquiring object models on the fly

As part of our efforts to build and continuously extend visual object models we investigated segmentation of objects from dynamic scenes, e.g. scenes where a human operator manipulates various objects. This includes building new models from scratch, maintaining and extending models over time and importantly handling of partial and complete occlusions. Reasoning about occlusions is crucial, as it allows to explain *why* tracking of a known object hypotheses suddenly starts to fail and underlies the decision to either update the model (as is necessary when tracking performance degrades due to appearance change such as out of plane rotation) or keep the model unchanged and register an occlusion event. To this end we maintain an over-complete set of object hypotheses capturing all the objects in the dynamical scene and select the most plausible interpretation based on a minimum description length (MDL) principle.

Concretely, our system uses affine model based motion clustering of interest points to create object hypotheses. Consistently moving interest points are clustered, thus building an initial model of an object hypothesis. In subsequent frames similar clusters confirm the evidence of an object and the model is extended by adding new interest points. This allows handling changes of appearances and rotation of objects to previously unseen views. An occlusion reasoning framework is used to track objects even un-

der full occlusion. A graph based spatio-temporal representation of multiple object hypotheses is maintained over time and thus the system is able to explain the scene even if objects are temporarily completely occluded. This leads to an over-complete set of object hypotheses. We use an MDL-based model selection framework to select a consistent interpretation for each image frame. The result of our approach is a set of object models created from all previously seen frames and the assumed location for each object including completely occluded objects. More details are reported in [21] (Annex 2.1). This work is a precursor to related work presented in DR.2.2 ([20], Annex 2.3).

In the above cases of the previous sections the system does not explicitly reason about its knowledge gaps. Instead knowledge gaps simply trigger the next visual process to fill in the missing information, based on task specifications (locating known objects in a room or on the table for the Dora and Dexter scenario respectively, or learning about object properties in the George scenario).

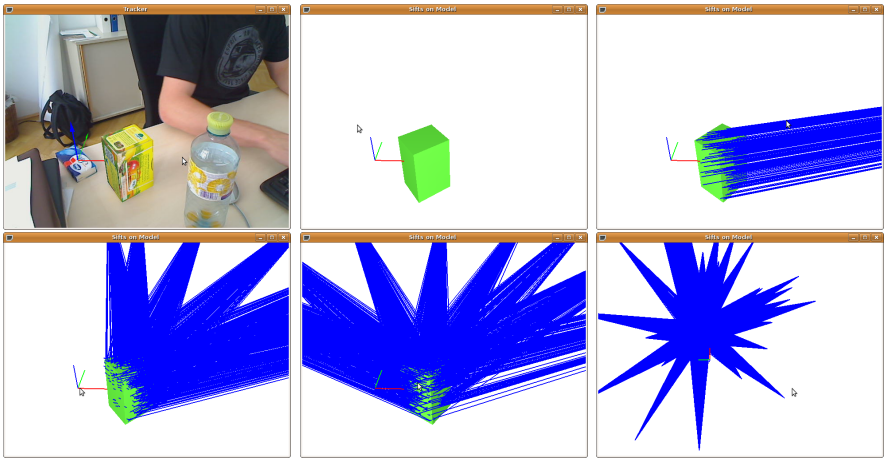


Figure 1: Completing the model: The tracked object in the scene (top left) and bundles of SIFT features with their view vectors (in blue) after acquiring more views of the object.

Extending our work on object recognition and tracking presented in DR.2.2 ([25, 18], Annexes 2.1 and 2.2 respectively) we recently started work on supporting more detailed and explicit reasoning about the completeness of object models and on using active exploration to complete object models. Object recognition is based on associating SIFT features with their position on the 3D object surface and performing robust RANSAC based 3D pose estimation. Models are built online while tracking the object based on the model acquired so far. New SIFT features extracted from the live image are mapped onto the 3D object surface. These features are also associated

with the view direction from which they were captured. So we know which object views are covered so far. Figure 1 illustrates how the object gets covered by more and more "bundles" of SIFT features with their associated view vectors. To assess the likelihood of detecting an object from a given view we place Gaussian distributions around the view vectors with a standard deviation of 30° which is derived from [14], which reports around 80% repeatability of detection for a single SIFT feature at 30° out of plane rotation. Summing over these followed by normalisation we obtain a PDF characterising the likelihood of detecting an object from a given view. This PDF represents the "density" of available object knowledge with respect to recognition and can be used to guide acquisition of new object views to areas not sufficiently covered.

1.2.4 Identifying stable grasps under uncertain object knowledge

In order to implement a full grasping cycle on a robot, on both known and unknown objects, it is important to equip the robot with the capability of reasoning about grasp stability. Before an action such as lifting is applied to an object, the robot should be capable of deciding whether a grasp applied to an object is stable enough to allow subsequent actions. The problem of grasp stability estimation has been studied extensively in the robotic literature. However, most of the methods are based on analytical approaches that assume a complete knowledge of object attributes such as shape, size, material properties, weight, etc. In addition, for exact stability estimation the knowledge of the contact points, that is, the exact pose of the object in the robot hand is required. In realistic scenarios, the uncertainty in the vision system or positioning of the robot arm with respect to the object will results in pose offsets that are not possible to measure with the sensory systems. In addition, for more dexterous hands such as the three-fingered hand used in our experiments, different velocities or opening angles of the fingers will result in movement of the object once the fingers are closed around it.

One of the goals of CogX is to implement a process of reflection that explains the experience arising from robot exploration. Thus, the robot can perform informed exploration strategies for execution of stable grasps using different robotic hands. One goal for this period of Task 2.10 was to concentrate on representations of gaps in object knowledge and manipulation skills. With this in mind, we have explored the possibility of using machine learning techniques for assessing grasp stability based on tactile data. The main contribution of the work is an investigation of probabilistic modelling for inferring grasp stability based on learning from examples. We want to classify a grasp as stable or unstable before applying further actions on it, e.g. lifting. The problem is important and cannot be solved by visual sensing which is typically used to execute an initial robot hand positioning

with respect to the object. The output of the classification system can trigger a re-grasping step if an unstable grasp is identified. Thus, this step can also be seen as an exploration strategy where the robot extends its knowledge through active interaction with the environment. An off-line learning process is implemented and used for reasoning about grasp stability for a three-fingered robotic hand using Hidden Markov models. To evaluate the proposed method, experiments are performed both in simulation and on a real robot system. The above work is reported in detail in [1] (Annex 2.2).

1.2.5 Modelling knowledge about physical object behaviour in simple manipulation tasks

In this section we focus on the use of context to model object motion caused by robotic manipulation, and on the detection of novelty in object behaviour. In order to predict the behaviour of a rigid object subjected to a simple manipulation, such as a push, we accept that uncertainty arises in several aspects of knowledge about the object, namely:

1. the precise trajectory realized by the object during manipulation
2. the object's identity (which determines intrinsic characteristics of the object, such as shape, size, weight, mass distribution, and surface frictional properties)
3. whether the object is novel or already known to the prediction system

Whereas the first two aspects are addressed by other tasks in Work Package 2, here we discuss representations and algorithms to detect a novel object by observing its motion when manipulated.

We base our work on the probabilistic model for predicting object motion that was summarized in DR 2.2 §1.2 on modular motor learning. Specifically we define a predictor to be the product of two probabilistic models or experts – a "global" and "local" expert [12] (DR 2.2 - Annex 2.6). When trained on simulated or real object trajectories, such a predictor describes the motion of a simple object (such as a "polyflap"), that is pushed along a table by a finger at the end of a robot arm.

Rather than have a single predictor handle all objects, we employ ideas on multiple models and contexts as described in the Modular Motor Learning theory of Wolpert and colleagues [27, 8]. However, the architecture we construct is a mixture of products of experts, instead of a variant of the mixture of experts model as in [27]. Thus the system learns several predictive models, one for each different context, where context can be taken to be object shape, weight, etc., or in general some configuration of the environment relevant to the manipulation task. We assume that there is a set of these context-based predictors, each trained on a particular context.

To perform context estimation during a test push with an (initially) unknown object, we employ a Bayesian model selection approach [11], using information about object motion obtained by visual tracking [19]. At any time point in the test trial, the most likely context is determined by how much likelihood each context-based predictor had allocated to the observed object trajectory up to that point.

To deal with novel contexts we introduce an extra "novelty" predictor that dominates when all the other models allocate a low probability to the observed trajectory. The novelty predictor assigns a constant probability to any observed object motion and acts as a probability threshold in the model selection procedure. The probabilistic scheme developed here is closely related to the one described in [28].

We have performed some preliminary experiments in which three types of polyflap are (separately) pushed along a table by a robotic finger. The contextual variable was the friction coefficient of the polyflap surface. For the simple case of two context-based predictors (each trained on one of the first two contexts), the context is estimated successfully, typically a second or two after the finger makes contact with the polyflap.

1.3 Relation to the state-of-the-art

Most attention operators such as the well known Itti & Koch saliency operator [10] aim to model human visual attention. In the context of this work we are more interested in attention as relevant to a robot's tasks, and here it is the 3D structure of the environment that provides the relevant cues. Other 3D attention operators such as [15, 6] use depth maps as simply another channel next to colour or texture, while we explicitly work in the 3D domain.

Most systems for segmenting objects from dynamic scenes and reasoning about occlusion are geared towards traffic scenes or person tracking [3, 16, 5], treating the scene essentially as 2D with layers of objects, while we handle full 3D scenes with out of plane rotations. Reasoning about object behaviour in [4] is based on image regions using trajectories and velocity information, while our reasoning is based on more abstract object behaviour to achieve a consistent scene interpretation even if objects are totally occluded. Some approaches such as [9] use rather simple colour based models, which is not sufficient once objects are rotated in 3D and have different colour and texture on various sides. Our approach is based on interest points and learned models explicitly encompass all trained views.

During the last few decades, there has been a significant amount of work reported in robotic object grasping, see [26] for a recent survey. Feedback from tactile sensors has been used to maximise the contact surface for removing a book from a bookshelf, [17]. In [22], the integration of force, visual and tactile feedback has been proposed for an application of opening

a sliding door. The main difference between the above approaches and the work presented in our work is that we concentrate on using the tactile sensors for assessment of grasp stability. Thus, rather than using the tactile data for control, we reason about the stability before starting to actively manipulate the object. There have been many examples of grasp planning demonstrated in simulation. Their commonality is the use of a strategy that relies on known object shape and/or pose. Modelling object shape with a number of primitives such as boxes and cylinders [13], or superquadrics [7] reduces the space of grasp hypotheses. The decision about the most suitable grasp is based on grasp quality measures given contact positions. However, these techniques do not deal with uncertainties that may arise in realistic scenarios. To our knowledge, the analysis of grasp stability using Hidden Markov models and tactile sensors presented in this paper has not been studied before.

2 Annexes

2.1 Prankl et al. "Motion guided learning of object models on the fly"

Bibliography Prankl, J.; Zillich, M.; Vincze, M.: "Motion guided learning of object models on the fly", 5th International Cognitive Vision Workshop (ICVW), St Louis, 2009.

Abstract Motivated by psychologists findings that infants already at the age of 4 months build a spatio-temporal representation of objects and perceive objects as a single entity because of coherent motion, we present a system which uses similar motion of interest points to guide the focus of attention for learning object models on the fly. The novelty of our system is to learn object models due to motion despite complex interactions of multiple objects. Consistently moving interest points are clustered, thus building the initial model of an object hypothesis. In the subsequent frames similar clusters confirm the evidence of an object and the model is extended by adding new interest points. In this way the system handles changes of appearances and rotating objects to previously unseen views. We represent objects in a star-shaped geometrical model of interest points using a codebook. A graph based spatio-temporal representation of multiple object hypotheses is maintained and thus the system is able to explain the scene even if objects are totally occluded. This representation is used for a consistent scene interpretation and to reason about possible object locations to compute a prior for object recognition.

Relation to WP We gradually build up object knowledge leading to a more and more complete interpretation of the scene and via occlusion events reason explicitly about incomplete knowledge and possible reasons. This constitutes a good example of self-extension (adding object models) aided by self-understanding (reasoning about possible disappearances of objects).

2.2 Bekiroglu et al. "Learning grasp stability based on tactile data and HMMs"

Bibliography Yasemin Bekiroglu, Danica Kragic and Ville Kyrki: "Learning grasp stability based on tactile data and HMMs", IEEE International Symposium on Robot and Human Interactive Communication (ROMAN), 2010, Viareggio, Italy.

Abstract In this paper, the problem of learning grasp stability in robotic object grasping based on tactile measurements is studied. Although grasp stability modeling and estimation has been studied for a long time, there are few robots today able of demonstrating extensive grasping skills. The main contribution of the work presented here is an investigation of probabilistic modeling for inferring grasp stability based on learning from examples. The main objective is classification of a grasp as stable or unstable before applying further actions on it, e.g. lifting. The problem is important and cannot be solved by visual sensing which is typically used to execute an initial robot hand positioning with respect to the object. The output of the classification system can trigger a regrasping step if an unstable grasp is identified. An off-line learning process is implemented and used for reasoning about grasp stability for a three-fingered robotic hand using Hidden Markov models. To evaluate the proposed method, experiments are performed both in simulation and on a real robot system.

Relation to WP This work deals with modelling knowledge about grasping of objects. In many practical cases object information (shape and/or pose) from vision alone is uncertain. We thus complement the systems knowledge about grasping with learned stability measures from tactile sensors. This allows the system to reason about grasp failures and replan grasping actions if necessary.

2.3 Zurek et al. "Using context to identify novelty during simple manipulation of rigid objects"

Bibliography Sebastian Zurek, Marek Kopicki, Rustam Stolkin, and Jeremy Wyatt: "Using context to identify novelty during simple manipulation of rigid objects", Technical Report, School of Computer Science, University of Birmingham, UK, 2010.

Abstract We adapt a model of human sensorimotor learning and control to the robotic domain. The modular motor learning theory of Wolpert and Kawato makes use of a set of motor controllers, in which each controller is suitable for one or a few contexts. Here a context is understood to be a configuration of the environment, such as object weight or shape. We apply this idea of context to predict the motion of a rigid object manipulated by a robotic finger. Given a trained set of predictors, Bayesian model selection is used to infer the context during a manipulation experiment. To detect novel contexts, a "novelty" predictor competes with the trained predictors in the model selection process. Preliminary results from an experimental trial, in which an object is pushed by a robotic finger, demonstrate how the estimate of context varies with time.

Relation to WP This work describes a probabilistic model of object behaviour for the case of robotic-finger manipulation of rigid objects. Based on the idea of context from modular motor learning theory, we describe how context-based predictors can be used to detect new contexts, i.e. gaps in object knowledge, by assessing the quality of their predictions.

References

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

[2] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *ICCV (1)*, pages 377–384, 1999.

[3] F. Brémond and M. Thonnat. Tracking multiple non-rigid objects in video sequences. *IEEE Transaction on Circuits and Systems for Video Technology Journal*, 8(5), September 1998.

[4] Rita Cucchiara, Massimo Piccardi, and Paola Mello. Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):119–130, June 2000.

[5] Ahmed M. Elgammal and Larry S. Davis. Probabilistic framework for segmenting people under occlusion. In *ICCV*, pages 145–152, 2001.

[6] S. Frintrop, E. Rome, A. Nuchter, and H. Surmann. A bimodal laser-based attention system. *J. of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance in Computer Vision*, 100(1-2):124–151, 2005.

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

[8] M. Haruno, D. M. Wolpert, and M. Kawato. MOSAIC model for sensorimotor learning and control. *Neural Computation*, 13:2201, 2001.

[9] Yan Huang and Irfan Essa. Tracking multiple objects through occlusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05)*, volume 2, pages 1051–1058, San Diego, CA, USA, June 2005.

[10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on PAMI*, 20(11):1254–1259, Nov 1998.

[11] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, Jun 1995.

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

[13] Danica Kragic, Andrew Miller, and Peter Allen. Real-time tracking meets online grasp planning. *IEEE International Conference on Robotics and Automation, ICRA'01*, pages 2460–2465, 2001.

[14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] A. Maki, P. Nordlund, and J.-O. Eklundh. A computational model of depth-based attention. In *Proceedings of the 13th ICPR*, volume 4, pages 734–739, Aug 1996.

[16] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, October 2000.

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

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

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

[20] J. Prankl, M. Zillich, B Leibe, and M. Vincze. Incremental Model Selection for Detection and Tracking of Planar Surfaces. In *BMVC*, 2010.

[21] J. Prankl, M. Zillich, and M. Vincze. Motion guided learning of object models on the fly. In *5th International Cognitive Vision Workshop (ICVW), St Louis*, 2009.

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

[23] A. Richtsfeld and M. Vincze. 3D Shape Detection for Mobile Robot Learning. Technical Report ACIN-TR-2009/1, Automation and Control Institute, Vienna University of Technology, Vienna, Austria, June 2009.

[24] A. Richtsfeld and M. Vincze. Basic Object Shape Detection and Tracking Using Perceptual Organization. Technical Report ACIN-TR-2009/2, Automation and Control Institute, Vienna University of Technology, Vienna, Austria, June 2009.

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

[26] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

[27] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.

[28] Jeremy Wyatt et al. Unifying representations of beliefs about beliefs and knowledge producing actions. CogX DR 1.2 §8, CogX Consortium, 2010.

[29] K. Zhou, M. Zillich, and M. Vincze. Reconstruction of Three Dimensional Spatial Clusters Using Monocular Camera. In *The 3IA International Conference on Computer Graphics and Artificial Intelligence*, Athens, Greece, 2009.

[30] Kai Zhou, Michael Zillich, Markus Vincze, Alen Vrecko, and Danijel Skocaj. Multi-model fitting using particle swarm optimization for 3d perception in robot vision. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010. submitted.

Motion Guided Learning of Object Models on the fly *

Johann Prankl, Michael Zillich, Markus Vincze

Automation and Control Institute
Vienna University of Technology, Austria
{*prankl,zillich,vincze*}*@acin.tuwien.ac.at*

Abstract. Motivated by psychologists' findings that infants already at the age of 4 months build a spatio-temporal representation of objects and perceive objects as a single entity because of coherent motion, we present a system which uses similar motion of interest points to guide the focus of attention for learning object models on the fly. The novelty of our system is to learn object models due to motion despite complex interactions of multiple objects. Consistently moving interest points are clustered, thus building the initial model of an object hypothesis. In the subsequent frames similar clusters confirm the evidence of an object and the model is extended by adding new interest points. In this way the system handles changes of appearances and rotating objects to previously unseen views. We represent objects in a star-shaped geometrical model of interest points using a codebook. A graph based spatio-temporal representation of multiple object hypotheses is maintained and thus the system is able to explain the scene even if objects are totally occluded. This representation is used for a consistent scene interpretation and to reason about possible object locations to compute a prior for object recognition.

Key words: Scene interpretation, Motion segmentation, Object recognition

* The work described in this article has been funded by the European Commission's Sixth and Seventh Framework Programmes under contract no. 6029427 (XPERO), no. 215821 (GRASP) and no. 215181 (CogX). The work was also supported by the Austrian Science Foundation under the grant #S9101 ("Cognitive Vision").

1 Introduction

One of the rising challenges is to endow our environment with capabilities to be sensitive and responsive to the presence of people. A necessary basic ability for such an artificial cognitive system (be it an ambient intelligent system or an autonomous robot) is to focus on the foreground and perceive objects as unity in contrast to the background. We aim to build a cognitive agent, which observes the environment and builds a spatio-temporal representation of object hypotheses. Our approach is motivated by psychologists' findings about infants that have shown that besides Gestalt principles and occlusion, motion is one of the most important cues to perceive object unity [1]. Gredebäck [2] and Spelke [3] have shown that infants already at the age of four months build a spatio-temporal representation of objects and accurately predict their reappearance after full occlusion.

Typical vision systems integrate low-level visual cues in a hierarchical fashion and extract relevant output from this bottom-up processing. Recent approaches try to establish feedback loops and combine different vision methods at various levels, but these methods also reach their limitations if dynamical scenes get crowded and objects get partly or even totally occluded. Our system fuses bottom-up visual processing with top-down reasoning to keep track of occluded objects and to learn appearances of objects that continuously change due to rotation or lighting. The system reasons about occlusion and hiding events and maintains an object hypothesis graph that is updated according to the visual input. We represent objects in a star-shaped geometrical model of interest points using a codebook. In case of a plausible object hypothesis from motion segmentation a learning event is triggered and the interest points of an existing object are updated or a new model is created, respectively.

We tested our system with a scenario where a human moves different objects, which interact several times, i.e., get occluded and reappear again. The goal is that the system learns object hypotheses because of coherent motion of interest points and keeps track of them even if they rotate to views which have never been seen before, or during periods of full occlusion.

The paper is structured as follows: After an overview of related work in the next section, the overall system is presented in Sec. 2. In Sec. 3, the object representation including the object hypothesis graph and the star-shaped codebook model are described. Then, the motion segmenter and the recognition component are described in Sec. 4 and Sec. 5. Finally, reasoning and hypotheses selection is described in Sec. 6 and results are shown in Sec. 7.

1.1 Related work

We present a system which integrates four basic functionalities, namely motion segmentation, tracking, object recognition and reasoning. For further readings about the first three we refer to the most relevant approaches described in [4], [5] and [6]. In what follows, we will review the state of the art regarding systems which include high level reasoning and occlusion handling.

There exist some occlusion reasoning systems for tracking or segmenting objects, mostly for traffic scenes or persons. The approaches in [7] and [8] use image regions for occlusion reasoning. A region may consist of one or more objects, the relative depth between objects is not considered. If occlusion happens, the system merges the affected regions into a new region. On the other hand a region is split, if the system is able to discriminate objects within this region. Elgammal and Davis [9] use a maximum likelihood estimation to estimate the best arrangement for people. To cope with the occlusion problem, Wu [10] proposes a dynamic Bayesian network with an extra hidden layer and in [11] tracking of multiple objects in dynamic scenes with long periods of occlusion is handled by detecting the visibility state of the objects. In case of occlusion, the whole assemble of objects is tracked.

Huang and Essa [12] present an approach for tracking a varying number of objects through temporally and spatially significant occlusions. The method is built on the idea of object permanence. They assume that a simple colour model is sufficient to describe each object in a video sequence, therefore they do not have to update their object models.

Bennett et al. [13] enhances tracking results of moving objects by reasoning about spatio-temporal constraints. The reasoning engine resolves error, ambiguity and occlusion to produce a most likely hypothesis, which is consistent with global spatio-temporal continuity constraints. However, the whole system does only bottom-up processing.

A way to incorporate knowledge into vision systems is to use a knowledge-based approach, e.g. [14] for an aerial image understanding system, and [15] for traffic monitoring applications. Matsuyama and Hwang [14] identified two types of knowledge in an image understanding system, that is, knowledge about objects and about analysis tools, and built a modular framework which integrates top-down and bottom-up reasoning. The system extracts various scene descriptions, and an evaluation function selects the most complex scene description from the database. The evaluation function is trivial and trusts the low-level vision output more than the reasoning output.

While reasoning in [15] is based on image regions using trajectories and velocity information, our reasoning is based on more abstract object behaviour to achieve a consistent scene interpretation even if objects are totally occluded.

2 System overview

Our system consists of four main parts (Fig. 1): the motion segmenter, the object detector, the reasoning component and the knowledge-base. The central role plays the reasoning component, which creates new object hypotheses triggered by the motion segmenter, maintains the hypothesis graph, predicts object locations to compute priors for the object detector and selects object hypotheses for a consistent interpretation of the current image. For each object of a specific image frame there exist several object hypotheses. Each hypothesis is linked to an object of the previous as well as to an object of the next frame. The reasoner

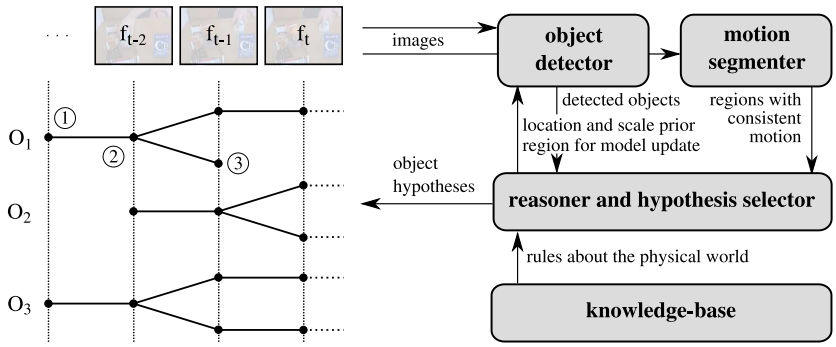


Fig. 1. System overview, including the hypothesis graph (left), the structure of the system and the communication between the different components (right).

either creates new object hypotheses of unseen motion clusters, or it creates object hypotheses including a copy of the object models of the last frames, or it creates object hypotheses with an updated model (see different nodes on the left in Fig. 1).

The result until here is an over-complete set of object hypotheses, that explains the same area of the image. To get a consistent interpretation of a particular frame a minimum description length (MDL) based selection framework is used and the best hypotheses mask weaker ones.

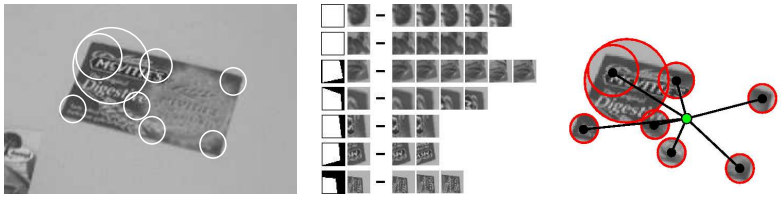
Before going into details with the different components, we describe the graph based representation of the object hypotheses.

3 Object representation

In contrast to classical object recognisers, which have an optimised model to recognise an object in one image, our approach works on image sequences and uses the history of the object hypotheses for modelling the object as well as for predicting the location in the next image. The object model is generated online and stored in the object hypothesis graph in a distributed manner.

3.1 Object hypothesis graph

The object hypothesis graph (see Fig. 1, left) is maintained by the reasoning component and stores all object locations and the models of the according views of all previous images up to the current frame. We use a star-shaped geometrical representation depicted in Fig. 2(c). Depending on the results of the object detector and the object segmenter the reasoner creates a new object hypothesis, i.e., it stores the interest points within a segmented region with respect to the centre of the region or the interest points are aligned with the stored model of the previous frame using the current detection result. Thus for each frame we have object hypotheses which are linked to the parent hypotheses of the



(a) (b) (c)

Fig. 2. Fig. 2(a) shows detected interest points and Fig. 2(b) the codebook representation, cluster means and occurrences (interest points). In Fig. 2(c) the star-shaped object representation is sketched.

previous frame and if there is a supporting segmentation for a detection result the current occurrences, i.e., the interest points within the segmented region, are stored. These occurrences are then used to generate a "small" model using the immediate previous frames optimised for tracking or – in case the object is lost – all previously seen occurrences are used to create a compact model optimised for object recognition. As proposed by Lowe [16] the Difference-of-Gaussian (DoG) operator and SIFT are used for detection and description of interest points.

3.2 Building compact object models

Our object model for recognition is inspired by the work of Leibe et al. [5], who proposed to build up a vocabulary ("codebook", see Fig. 2(a) and 2(b)) of interest points and to compose a geometric structure out of this vocabulary. We extended this approach with a geometric pruning algorithm to get a more compact model and thus speed up object detection.

The first step is to create a codebook for each object. Therefore links of current object hypotheses are traced back to the parent objects and the according descriptors of the occurrences are clustered following the RNN-algorithm as described in [5]. The RNN-algorithm is an agglomerative clustering algorithm which successively merges local descriptors until a cut-off threshold is reached. Thus this algorithm automatically determines the number of clusters while ensuring the cluster compactness. The next step is to assign the geometric locations to each cluster mean. Hence each codebook entry can vote for several object locations described in detail in Sec. 5. We use sequences of images thus a lot of similar occurrences build a codebook entry which offers the possibility for a statistical analysis to prune unreliable occurrences. In a post processing step the codebook is optimised to speed up the object detection, therefore we apply a geometric hashing for each codebook entry, in which hash bins must have at least two entries otherwise the according occurrence is deleted. Then the codebook is examined and all entries with less than two occurrences are deleted.

Summarised, separate codebooks are created for each object including occurrences of at least 3 frames for tracking and occurrences of all previous frames if an object gets lost. Clustering and geometric pruning is used to build a compact

object model including only reliable occurrences. These object models are then used for recognition described in Sec. 5.

4 Motion segmentation

The whole system is triggered by the motion segmentation component. We do not rely on a perfect segmentation of moving objects, but rather take care to achieve robustness later due to the cognitive component described in Sec. 6. Consequently, we just use a fast clustering of interest points depending on their affine motion. Our approach is inspired by the work of Pundlik et al. [4], who presented a real-time incremental approach to motion segmentation operating on sparse feature points. In contrast to Pundlik, who randomly selects interest points and uses an incremental growing algorithm, we use a 2-dimensional histogram of the length of the motion vectors and the motion direction to obtain good initial pre-clusters. Then a splitting algorithm and an outlier detection follows and theses clusters are merged depending on similar affine motion.

(a) (b) (c)

Fig. 3. Grouped motion vectors of interest points are shown with identical colours; different colours mean different clusters. Each subfigure depicts the result of different processing steps. Fig. 3(a) shows the result of grouping according to similar length and direction of the motion vectors using a 2-dimensional histogram. Fig. 3(b) is the result after examination of the neighbouring motion vectors using a delaunay triangulation and after affine outlier detection. In Fig. 3(c) the result of Fig. 3(b) is used to initialise a merging algorithm which combines clusters depending on their affine motion.

In detail: the first step is to examine the 2D-motion histogram. Therefore we search for all local maxima, that is we look for histogram bins which are surrounded by bins with a lower number of entries. Starting from the local maxima all neighbouring bins are clustered until a saddle bin is found. The result can be seen in Fig. 3(a). The next step is to split large clusters in case of intersecting convex hulls of other clusters. Therefore we use a delaunay tree to create a location neighbourhood graph of all interest points detected in the image. The splitting criterion prohibits intersections of two clusters and thus

substitutes a cluster with two new ones if they have no connection within the delaunay tree. After an affine outlier detection using a Least Median of Squares implementation, publicly available at FORTH [17] (see Fig. 3(b)), the clusters are again merged if the affine error is lower than the maximal error would be if they stayed separated. Thus the merging criterion results in

$$C_m = C_i \cup C_j \text{ for } e_m < \max(e_i, e_j) \tag{1}$$

In (1), C_i and C_j are two clusters, which are tested for similar affine motion and C_m denotes the merged cluster. e stands for the affine errors of the clusters. Additionally we can adjust a chaining parameter to cluster only features which are tracked for more than two frames and are thus considered as more stable.

Fig. 3 shows the results of all three main steps. It can be seen that the three outliers on the hand are filtered as well as the mismatch on the keypad of the telephone. The final motion clusters are handed over to the reasoning component which initialises a new object hypothesis or adds the features to an existing object. This is explained in detail in Sec. 6, after the object recogniser is described.

5 Object detection

The same interest points (DoG-operator and SIFT-descriptor [16]) used for the motion segmentation just described, are also used for object recognition. The detected interest points are matched with the codebook and activated codebook entries vote for an object centre.

Consistent votes are accumulated in the Hough accumulator array. We use a three dimensional space where occurrences of activated codebook entries vote for an object location $\mathbf{x}_v = (x_v, y_v)$ and a scale s_v:

$$s_v = \frac{s_i}{s_{occ}}, \tag{2}$$

$$\mathbf{x}_v = \mathbf{R}\mathbf{x}_{occ}s_v + \mathbf{x}_i. \tag{3}$$

In (2), s_i is the scale of the detected interest point in the current image and s_{occ} denotes the scale of the occurrence in the learning image, respectively. In (3) $\mathbf{x}_i$ is the location of the detected interest point in the current image, $\mathbf{x}_{occ}$ denotes the location of the object centre with respect to an occurrence of the model and $\mathbf{R}$ stands for the matrix that describes the rotation from model to image orientation of the interest point.

Once all matched interest points have voted, the Hough accumulator array is used to find the most promising object hypotheses. The probabilistic votes in each Hough bin i are summed up and – starting with the best hypothesis, i.e., the largest bin – the object location is refined. This is done in a mean shift like procedure, for which the neighbouring bins are examined for contributing votes. This handles the typical boundary effect of Hough voting schemas.

The result of the mean shift refinement is a cluster of interest points, that consistently vote for an object location. This cluster is used to compute an affine

homography H_{aff}, for which we use the Least Median of Squares implementation already mentioned in Sec. 4. H_{aff} is further used to project the model boundary to the current frame. The projected boundary is not only used for visualisation but also for interest point statistics and for computation of the confidence value

$$c(o|m, f_t) = -\kappa_1 + (1 - \kappa_2) \cdot \frac{n_{matched}}{n_{detected}} + \kappa_2 \cdot \frac{s_{matched}}{n_{detected}} \tag{4}$$

of an object o for a given frame f_t and an object model m. $n_{matched}$ are the matched interest points and $n_{detected}$ are the number of the detected interest points located within the boundary projected to the current frame. $s_{matched}$ is the sum of the weights

$$w = p_m p_{occ} = \frac{1}{n_m \cdot n_{occ}} \tag{5}$$

of all matched interest points with p_m and p_{occ} denoting the probabilities of the match and the occurrence in the model, respectively. n_{occ} is the number of occurrences of the specific object model of the activated codebook entry and n_m is the number of activated entries of the interest point. κ_1 and κ_2 are two constants which weight the different factors.

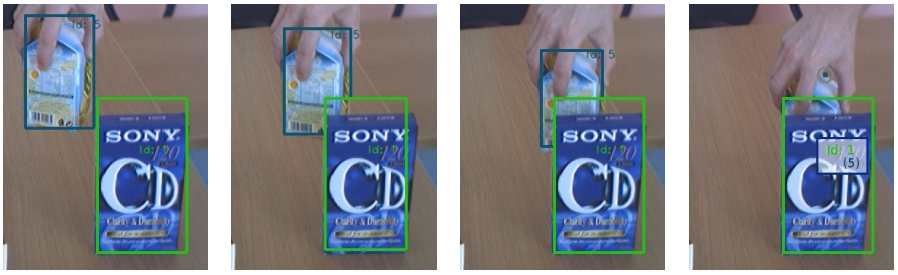
6 Reasoning and hypotheses selection

The central role plays the reasoning component, it predicts object locations both for the case of tracking and for the case of total occlusion. It creates new object models or updates existing models depending on coherent segmentation and detection results and it selects objects from an over-complete set to get a consistent scene interpretation. The following sections describe the different functionalities of the reasoner starting with the occlusion analysis.

6.1 Occlusion analysis

We aim to get a consistent interpretation of an image sequence thus it is necessary to predict objects even if they are totally occluded. Therefore we developed an event based occlusion analysis schema. If an object gets lost the past, the current and the predicted object locations in the future are examined for possible occluders. Therefore for each location the overlap of the projected object boundary with the other visible object hypotheses is computed and if they overlap the visible object gets an occlusion vote. The voting is done for all past and future object locations which are within a maximum distance of half the object size. After the visible objects accumulated the votes, the ID of the occluded object is assigned to the visible one with the most votes and to all other which got more than 80% of the maximum. It turned out that this voting schema is more reliable than only looking at the position of disappearance because in case of partial occlusion our model updating algorithm tends to shrink the estimated object boundary to the visible part of the object.

Fig. 4 shows an occlusion event, the correct depth ordering which is estimated from the confidence value (cp. Sec. 6.2), and the link of an occluded object to the occluder (indicated by an object ID within the brackets).



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

Fig. 4. Occlusion event including correct depth ordering and an occluded object linked to the occluder

6.2 Confidence value for tracking using a location and a scale prior

For tracking the objects we use a constant velocity assumption therefore the affine homography H_{inc} between two frames is computed for each object. Then the assumed location and scale is computed for objects of the current frame and the confidence value is extended to

$$c_{track}(o|m, f_t) = c(o|m, f_t) + \kappa_3 \cdot \log\, p(o_{f_t}|o_{f_{t-x}}) + \kappa_4 \cdot \log\, s(o_{f_t}|o_{f_{t-x}}) \quad (6)$$

where $p(o_{f_t}|o_{f_{t-x}})$ and $s(o_{f_t}|o_{f_{t-x}})$ stand for the location and the scale prior and κ_3 and κ_4 are further constants to weight the priors. We model the priors using a Gaussian around the predicted location and the last scale. In case of occlusion the location prior is extended and surrounds the whole boundary of the occluder. Thus reappearing objects are accepted near the occluder and at the last seen location (see Fig. 5). Then the objects are sorted according to the tracking confidence value and added to the hypothesis tree.

Fig. 5. Probability map for one specific object computed using the predicted location and the result of the occlusion analysis.

6.3 Maintenance of the object models

The next step is to update existing object hypotheses. Therefore we compute an overlap matrix which describes the support of segmented regions and detected object hypotheses. We define the support

$$support_{i,j} = \frac{A_{seg} \cap A_{det}}{A_{seg} \cup A_{det}}. \tag{7}$$

where the support of a segmented region r_{seg} for a detected object hypothesis o_{det} is the ratio of the intersection and the union of the segmented area A_{seg} and the area of the detected object hypothesis A_{det}. For our experiments we used a *winner takes all* updating strategy, meaning that the detection result with the highest tracking confidence value is updated if the support is larger than a threshold t_{update}. Additionally we use a second threshold t_{new} for creating a new object hypothesis. If a segmented region does not support any detection result more than t_{new} a new hypothesis is created. Depending on the detection results and these two thresholds an over-complete set of object hypotheses is created from which hypotheses explaining the scene in a consistent way are selected.

6.4 Hypotheses selection

Our hypotheses selection framework was introduced in [18] and adapted by [5]. The idea is that the same data set cannot be occupied by more than one object and that the models cannot be fitted sequentially. Thus an over-complete set of hypotheses is generated and the best subset is chosen using a minimum description length criterion.

In our case the data set consists of the interest points and each interest point can only be assigned to one object model. Hence, overlapping models compete for interest points which is represented by the interaction costs q_{ij}. In contrast q_{ii} represents the merit term of an object hypothesis. Finding the optimal set of models leads to a Quadratic Boolean Problem (QBP)

$$\max_{n} \mathbf{n}^T Q \mathbf{n} \ , \ Q = \begin{bmatrix} q_{11} & \cdots & q_{1N} \\ \vdots & \ddots & \vdots \\ q_{N1} & \cdots & q_{NN} \end{bmatrix} \tag{8}$$

where $\mathbf{n}$= $[n_1, n_2, \cdots, n_N]$ stands for the indicator vector with $n_i = 1$ if an object hypothesis is selected and $n_i = 0$ otherwise. Q is the interaction matrix with the diagonal elements $q_{ii} = c_{track}(o|m, f_t)$ and the off-diagonal elements

$$q_{ij} = -\frac{1}{n_{o,weak}} \cdot ((1 - \kappa_2) \cdot n_{overlap} + \kappa_2 \cdot s_{overlap}) \tag{9}$$

where $n_{o,weak}$ is the number of interest points within the projected boundary to the current frame of the weaker hypothesis, i.e. with the lower confidence value, $n_{overlap}$ stands for the number of interest points which are shared by both objects and $s_{overlap}$ is the sum of the weights of all shared interest points (cp. Eq. 5).

6.5 Pruning of weak object hypotheses

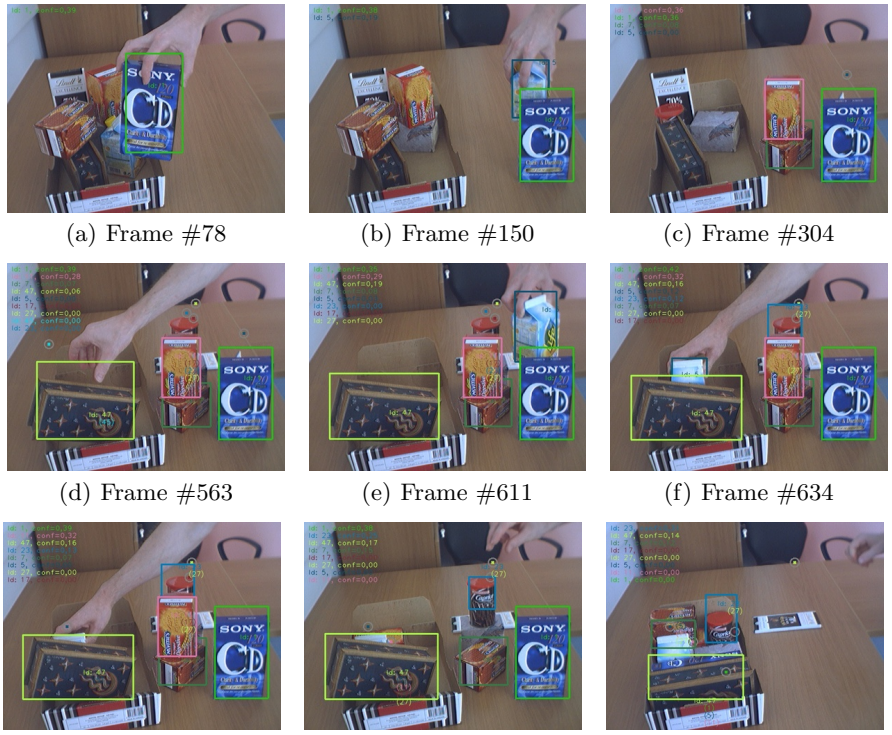
In case of a weak support of a segmentation and a detection our system generates additional object hypotheses. Continually extending the object hypothesis graph would lead to an intraceable system. Thus we introduced a *lifetime* of object hypotheses and delete models if they are not continuously updated. Motivated by the human brain, which has an exponential forgetting curve – discovered by Hermann Ebbinghaus in 1885 – we introduced an exponential lifetime

$$t_{life} = \frac{n_{seg}}{n_{life}} \cdot e^{\frac{n_{seg}}{c_{oblivion}}}. \tag{10}$$

where n_{seg} is the number of supports of a segmentation for an object, n_{life} is the number of frames since the object hypothesis was created and $c_{oblivion}$ stands for a constant to care for inaction time. Thus object hypotheses are only maintained if $t_{life} > 1$, otherwise the object model is deleted. This leads to a linear characteristics at the time when the hypothesis is created. If the object is supported by a segmentation more often it will be stored almost forever.

7 Results

We processed six video sequences to test our system. In the following, we present three sequences, which show the strengths as well as the weaknesses. The system has to detect object hypotheses because of consistent moving interest points, interpret the sequence correctly including hypotheses for totally occluded objects and build object models with all seen views. In our first video sequence, called *Sorting the Shopping Basket* we arranged typical household articles in a crowded manner in a box. Then a person empties the box, resorts the articles and places them into the box again. The sequence shows a lot of complex interactions and it is taken at a low framerate (objects move more than 40 pixels between two frames) to show that our system can handle motion blur and that it is not bounded to a strict tracking assumption, but rather selects the best interpretation which is currently available. Fig. 6 shows selected frames of the sequence. Currently available object models are depicted with bounding boxes and the according IDs and confidence values are displayed at the upper left area of each image. In Fig. 6(a) the first object is grasped and because of the motion an object hypothesis with ID 1 is generated. The next Fig. 6(b) shows the second object (ID 5) which moves behind the first object. Correct occlusion assignment and the last detected location are indicated with the ID within brackets under the occluder ID and with a coloured dot surrounded by a grey circle. During complex actions sometimes "hallucinated" object hypotheses are created (Fig. 6(d) object ID 45) which are not confirmed and thus deleted in the following frames. In Fig. 6(e) the object with ID 5 re-appears. In this frame all eight correctly learned object models are listed in the upper left area of the image. After some interactions shown in Fig. 6(f), 6(g) and 6(h) the sorted box with correct occlusion assignment is depicted in Fig. 6(i). Only the chocolate bar (ID 27) is



(a) Frame #78 (b) Frame #150 (c) Frame #304

(d) Frame #563 (e) Frame #611 (f) Frame #634

(g) Frame #640 (h) Frame #707 (i) Frame #902

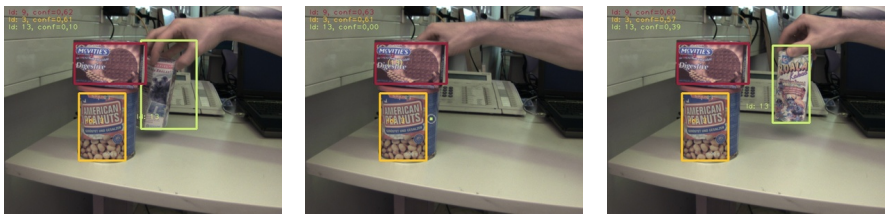
Fig. 6. Selected frames of the video named *Sorting the Shopping Basket*, indicating the complex interactions. Bounding boxes of learned objects are shown with different colours and the according IDs and confidence values of all currently available models are depicted at the upper left area of each image. If an object is lost the last position is depicted with a coloured dot surrounded by a grey circle and the ID of the occluded object is displayed under the occluder ID within brackets.

not recognised again, because of a too drastic change of the size and a too large rotation while it was occluded (i.e., no model was generated of this view before).

The second sequence depicted in Fig. 7 contains three foreground objects. One of the objects (ID 13) is rotated to different views. Then this object moves behind the other two and – triggered by the occlusion event – a model of all views, which have been shown before, is computed. During full occlusion the object is rotated and re-appears with a view shown at the beginning. It is correctly recognised again in Frame #850 (Fig. 7(f)).

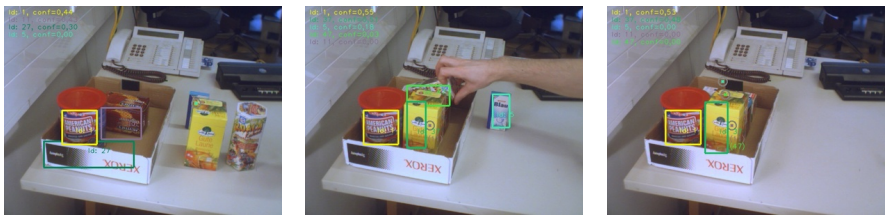
In Fig. 8 another sequence with household articles is shown. Despite the correctly learned object models two errors occurred. The first one is that the model of the xerox box (ID 27) has disappeared. This object hypothesis is not confirmed often enough during tracking and thus it has been deleted due to our forgetting curve. The second error is that the occluded object with ID 5 is not linked to the occluder 37. Because of a rotation in depth during occlusion

(a) Frame #578 (b) Frame #640 (c) Frame #676



(d) Frame #772 (e) Frame #810 (f) Frame #850

Fig. 7. Part of a 900 frames long video which indicates the learning of an object model including the history of the object. The model of object 13 is learned while rotating to completely different views. Then it is moved behind object 3 and 9. During full occlusion the object is rotated and appears again with a view learned at the beginning.



(a) Frame #1820 (b) Frame #2175 (c) Frame #2460

Fig. 8. Three images of a 2590 frames long video are depicted showing two possible errors.

the prediction was wrong and thus object 5 did not get in contact with the occluder 37.

8 Conclusion

In this paper we presented a system that uses an affine model based motion clustering of interest points to create object hypotheses. If the hypotheses are confirmed in the following frames more complex object models are created. An occlusion reasoning framework is used to track objects even under full occlusion. This leads to an over-complete set of object hypotheses. We use an MDL-based model selection framework to select a consistent interpretation for each image frame. The result of our approach is a set of object models created from all previously seen frames and the assumed location for each object including completely occluded objects.

References

1. Smith, W.C., Johnson, S.P., Spelke, E.S.: Motion and edge sensitivity in perception of object unity. Cognitive Psychology **46**(1) (2003) 31 – 64
2. Gredebäck, G.: Infants Knowledge of Occluded Objects: Evidence of Early Spatiotemporal Representation. Number Dissertation, ISBN 91-554-5898-X. Acta Universitatis Upsaliensis; Faculty of Social Sciences (2004)
3. Spelke, E.S., von Hofsten, C.: Predictive reaching for occluded objects by 6-month-old infants. In: Journal of Cognition and Development. Volume 2(3)., Lawrence Erlaum Associates, Inc. (2001) 261–281
4. Pundlik, S., Birchfield, S.: Real-time motion segmentation of sparse feature points at any speed. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **38**(3) (June 2008) 731–742
5. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. Int. J. Comput. Vision **77**(1-3) (2008) 259–289
6. Leibe, B., Schindler, K., Cornelis, N., Gool, L.V.: Coupled object detection and tracking from static cameras and moving vehicles. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10) (2008) 1683–1698
7. Brémond, F., Thonnat, M.: Tracking multiple non-rigid objects in video sequences. IEEE Transaction on Circuits and Systems for Video Technology Journal **8**(5) (September 1998)
8. McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H.: Tracking groups of people. Computer Vision and Image Understanding **80**(1) (October 2000) 42–56
9. Elgammal, A.M., Davis, L.S.: Probabilistic framework for segmenting people under occlusion. In: ICCV. (2001) 145–152
10. Wu, Y., Yu, T., Hua, G.: Tracking appearances with occlusions. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2003) 789
11. Yang, T., Li, S.Z., Pan, Q., Li, J.: Real-time multiple objects tracking with occlusion handling in dynamic scenes. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1, Washington, DC, USA, IEEE Computer Society (2005) 970–975
12. Huang, Y., Essa, I.: Tracking multiple objects through occlusion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05). Volume 2., San Diego, CA, USA (June 2005) 1051–1058
13. Bennett, B., Magee, D.R., Cohn, A.G., Hogg, D.C.: Using spatio-temporal continuity constraints to enhance visual tracking of moving objects. In: ECAI-04. (2004) 922–926
14. Matsuyama, T., Hwang, V.S.: SIGMA: A Knowledge-Based Aerial Image Understanding System. Perseus Publishing (1990)
15. Cucchiara, R., Piccardi, M., Mello, P.: Image analysis and rule-based reasoning for a traffic monitoring system. IEEE Transactions on Intelligent Transportation Systems **1**(2) (June 2000) 119–130
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
17. Lourakis, M.: homest: A c/c++ library for robust, non-linear homography estimation. [web page] `http://www.ics.forth.gr/~lourakis/homest/` (Jul. 2006) [Accessed on 20 Jul. 2006.].
18. Leonardis, A., Gupta, A., Bajcsy, R.: Segmentation of range images as the search for geometric parametric models. Int. J. Comput. Vision **14**(3) (1995) 253–277

Learning grasp stability based on tactile data and HMMs

Yasemin Bekiroglu, Danica Kragic and Ville Kyrki

Abstract— **In this paper, the problem of learning grasp stability in robotic object grasping based on tactile measurements is studied. Although grasp stability modeling and estimation has been studied for a long time, there are few robots today able of demonstrating extensive grasping skills. The main contribution of the work presented here is an investigation of probabilistic modeling for inferring grasp stability based on learning from examples. The main objective is classification of a grasp as stable or unstable before applying further actions on it, e.g. lifting. The problem is important and cannot be solved by visual sensing which is typically used to execute an initial robot hand positioning with respect to the object. The output of the classification system can trigger a regrasping step if an unstable grasp is identified. An off-line learning process is implemented and used for reasoning about grasp stability for a three-fingered robotic hand using Hidden Markov models. To evaluate the proposed method, experiments are performed both in simulation and on a real robot system.**

I. Introduction

For a general purpose service robot, operating in an industrial or a domestic environment, object grasping and manipulation skills are a necessity. Most of the todays robot systems however, demonstrate only limited object grasping and manipulation capabilities. The classical work in robotic grasping rests on an assumption that the object parameters such as pose, shape, weight and material properties are known. If precise knowledge of these is available, grasp stability estimation using analytical approaches is often enough for successful grasp execution. However, in unstructured environments the information is usually uncertain, which presents a challenge for the current systems.

To cope with the uncertainty, one can rely on sensory information for closed loop control, [1]. For grasping and manipulation, shape and pose of an object are important inputs to the control loop. However, the accuracy of vision is limited and small errors in object pose can cause failures. These failures are difficult to prevent at the grasp planning stage and need to be taken into account once the contact with the object has been made. Visual servoing approaches [2], [3] can solve these problems only to a certain extent since they commonly need a desired pose with respect to the object to be defined beforehand which is impossible for unknown objects. While the tactile and force sensors can be used to reduce the uncertainty upon contact, a grasp may fail even when all fingers have adequate contact forces. The major issue is that for unknown objects, grasps need to be evaluated from data the robot can extract on-line. Besides the incomplete information about the environment and the objects, there is also a lack of generalizable quality measures for grasp stability assessment under uncertainty.

We present a learning system that infers grasp stability based on tactile sensors. If an unstable grasp is detected, a regrasping step can be initialized before, for example, lifting the object. To achieve a good generalization performance, machine learning approaches typically require large amount of training data. As a solution to the problem of acquiring enough training data, we propose to first simulate the grasping process. Then, we evaluate the feasibility of the approach both on simulated and real data. We have implemented a time-series analysis based on a sequence of tactile measurements with the purpose of investigating the effect of the dynamic process of grasp execution on grasp stability. The results show that the idea of exploiting a learning approach is feasible. The additional contribution of the work is a publicly available database of the experimental sequences, [4].

The paper is organized as follows. Related work is reviewed in Section II and the notation summarized in Section III. Then, Section IV introduces the time-series recognition approach using Hidden Markov models. In Section V, the process of generation of the training data is described. Section VI presents the experimental results. Finally, we conclude and discuss directions for future research in Section VII.

II. Related Work

During the last few decades, there has been a significant amount of work reported in robotic object grasping, see [5] for a recent survey. In our previous work, we have integrated vision based object recognition and tactile sensing for closed loop grasp control,[1]. Regarding vision based approaches, a number of proposed solutions rely on object recognition and/or shape registration. This commonly requires a database of objects or shapes, as for example in [6], or even of objects combined with grasps, as presented in [7].

The feedback from tactile sensors has been used to maximize the contact surface for removing a book from a bookshelf, [8]. In [9], the integration of force, visual and tactile feedback has been proposed for an application of opening a sliding door. The main difference between the above approaches and the work presented here is that we concentrate on using the tactile sensors for assessment

Y. Bekiroglu and D. Kragic are with the Centre for Autonomous Systems and Computational Vision and Active Perception Lab, School of Computer Science and Communication, KTH, Stockholm, Sweden. V. Kyrki is with the Department of Information Technology, Lappeenranta University of Technology, Finland. yaseminb,danik@csc.kth.se, kyrki@lut.fi

This work was supported by EU through the project CogX, IST-FP6-IP-027657, and GRASP, IST-FP7-IP-215821 and the Swedish Foundation for Strategic Research.

of grasp stability. Thus, rather than using the tactile data for control, we reason about the stability before starting to actively manipulate the object.

There have been many examples of grasp planning demonstrated in simulation. Their commonality is the use of a strategy that relies on known object shape and/or pose. Modeling object shape with a number of primitives such as boxes and cylinders [10], or superquadrics [11] reduces the space of grasp hypotheses. The decision about the most suitable grasp is based on grasp quality measures given contact positions. However, these techniques do not deal with uncertainties that may arise in realistic scenarios.

The work of integrating learning with grasping is also related to understanding human grasping strategies. In [12], we have demonstrated how a robot system can learn grasping strategies from human demonstration using a grasp experience database. The human grasp was recognized with the help of a magnetic tracking system and mapped to the kinematics of the robot hand using a predefined lookup-table. More recent work uses vision based grasp recognition in a learning-by-demonstration framework, [13]. More recent learning approaches using tactile sensors are focused on either determining the shape properties of objects [14] or object recognition [15], [16].

To our knowledge, the analysis of grasp stability using Hidden Markov models and tactile sensors presented in this paper has not been studied before.

III. Feature Representation

As mentioned, the goal of the paper is to show how grasp stability can be assessed based on temporal sequences of tactile data using Hidden Markov models. The basic idea is to position a hand with respect to the objects so that a grasp can be obtained by closing the fingers. A robot hand is equipped with two-dimensional tactile patches at the fingertips. Tactile measurements are recorded from the moment the first contact with the object is obtained and until there is not change in the measurements detected. The whole measurements sequence is denoted $x_1^i, \ldots, x_{T_i}^i$. For comparison reasons, we will also present results of one-shot classification based only on a single tactile measurements, $x_{T_i}^i$ taken at the end of a grasping sequence. The data is generated both in simulation and on a real hardware and it will be presented in more detail later on. The notation used in this paper is as follows:

- $x_t^i = [M_f^{i,t} j_r^{i,t}]$ is the observation at time instant t given i-th sequence; f denotes the number of tactile sensors and r denotes the number of joints of the robot hand.
- $o_i = [x_t^i], t = 1, ..., T_i$ is an observation sequence.
- $D = [o_i], i = 1, .., N$ denotes a data set with N observation sequences.
- $M_f^{i,t} = m_{p,q}^{H_f^{i,t}}$ are the moment features extracted from the tactile readings on the sensor f at time instant t given i-th sequence. Details about the extraction of these are given later in this section.
- $j_r^{i,t}$ are hand joint angles at time instant t given i-th sequence.

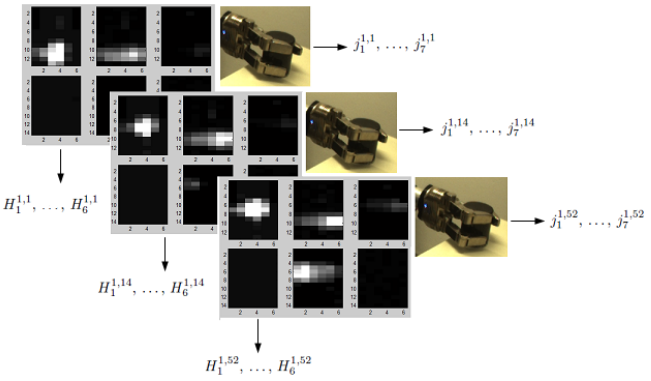


Fig. 1. An example grasping sequence of a cylinder and the corresponding tactile measurements.

- $H_f^{i,t}$ are the tactile readings collected from the sensor f at time instant t given i-th sequence.

The acquired data consists thus of tactile readings H_f^i and joint angles of the hand j_r. In simulation, the data originates from three tactile sensors: one per finger given the Schunk Dextrous Hand (SDH). Each sensor produces 12×6 tactile measurements and there are additionally seven parameters representing the pose of the hand given the joint angles. For the real world data, we used two different robot hands. For the Schunk Dextrous Hand, we store $3 \times (14 \times 6)$ readings on proximal and $3 \times (13 \times 6)$ on distal sensors. The second robot is a parallel 2-fingered gripper that equipped with the same type of tactile sensors that thus delivers $2 \times (14 \times 6)$ readings. Example images from the sensors are shown in Figure 1. The tactile images in the figure represent a stable grasp of a cylinder.

The tactile data is relatively high dimensionality and to some extent redundant. Therefore, we start by representing the acquired data as features. Here, we borrow some ideas from image processing and consider the two-dimensional tactile patches as images. In order to achieve an invariant representation as well as dimensionality reduction, we employ image moments as a suitable representation. The general parameterization of image moments is given by

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x, y) \tag{1}$$

where p and q represent the order of the moment, x and y represent the horizontal and vertical position on the tactile patch, and $f(x, y)$ the measured contact. We compute moments up to order two, $(p + q) \in \{0, 1, 2\}$, for each finger separately. These then correspond to the total pressure and the distribution of the pressure in the horizontal and vertical direction.

We normalize the zeroth order moment by calculating the average pressure $m_{0,0}/area$. First and second order moments are included in the feature vector according to Equation 1. Two additional features are computed for each tactile sensor/finger: the size of the contact area ($area$) and the center of the contact $(\frac{m_{1,0}}{m_{0,0}}, \frac{m_{0,1}}{m_{0,0}})$. Thus, there are in total

nine features for each sensor resulting in a feature vector $\theta_t \in \mathbb{R}^{9s}$ where s is the number of sensors for each hand: $s = 3$ in the case of the SDH.

Normalizing the feature vector is a common step in machine learning methods. In our case, moment features and finger joint angles are normalized to zero-mean and unit standard deviation. Normalization parameters are calculated from the training data and then used to normalize the testing sequences.

IV. Theoretical framework

This section presents the basics of the Hidden Markov models (HMMs) [17] and their application in our work. We train two HMMs: one that represents stable grasps and one that represents unstable ones. Recognition is then performed using the classical forward procedure: evaluating the likelihood given both models and the final decision is based on maximizing the estimated likelihood.

For the HMM, we use the classical notation $\lambda = (\pi, A, B)$ where π denotes the initial probability distribution, A is the transition probability matrix

$$A = a_{ij} = P(S_{t+1} = j | S_t = i), i = 1 \dots N, j = 1 \dots N \tag{2}$$

and B defines output (observation) probability distributions

$$b_j(x) = f_{X_t|S_t}(x|j) \tag{3}$$

Here, $X_t = x$ represents a feature vector for any given state $S_t = j$. The structure of an HMM can be ergodic or left-to-right, which determines the structure of A. In the following, we present and evaluate both of these models.

A. Modeling Observations

The estimation of the HMM model parameters is based on the classical Baum-Welch procedure. The output probability distributions are modeled using Gaussian Mixture Models (GMMs):

$$f_X(x) = \sum_{k=1}^{K} w_k \frac{1}{2\pi^{L/2}\sqrt{|C_k|}} e^{-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)} \tag{4}$$

where $\sum_{k=1}^{K} w_k = 1$, μ_k is the mean vector and C_k is the covariance matrix for the k-th mixture component. The unknown parameters $\theta = (w_k, \mu_k, C_k : k = 1...K)$ are estimated from the training sequences $o = (x_1, ...x_T)$.

Initial estimates of the observation densities in (Eq. 4) affect convergence of the reestimation formulas. Depending on the structure of the HMM, we employ different initialization methods for the parameters of the observation densities. The two initialization procedures are denoted $Init_1$ and $Init_2$:

- $Init_1$: For an ergodic HMM, observations are clustered using k-means. Here, k is equal to the number of states in the HMM and each cluster is modeled with a GMM using standard Expectation Maximization. Initial parameters for the GMMs are found in the standard fashion using the k-means algorithm.

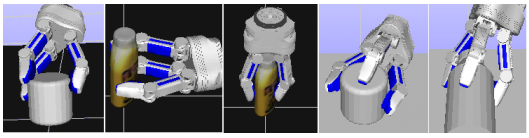


Fig. 2. Example grasps on different objects from five simulated datasets denoted (D_{S_1}), (D_{S_2}), (D_{S_3}), (D_{S_4}), (D_{S_5}) in the text.

- $Init_2$: For a left-to-right HMM, each observation sequence is divided temporally into equal length subsequences. Then, each GMM is estimated from the collection of corresponding subsequences. Thus, the GMMs represent the temporal evolution of the observations. Initial parameters for the GMM estimation are found identically to $Init_1$.

V. Data generation

The data was generated both in simulation environment and using real robotic hands. Both in real and simulated setups, a grasping sequence is recorded from tactile readings and corresponding joint configurations from the first contact with an object is made until a static state is achieved. After placing the hand in front of an object in a fully open position, the fingers are controlled to a closing position with equal velocity. By a static state, we consider a state when the tactile sensors do not report any change or fully closed hand configuration has been reached. The latter can occur only in the case the object was dropped or moved during the hand closing step.

The simulated data was generated to investigate two aspects of grasp stability recognition: shape specific and shape independent stability recognition. For shape specific recognition, the grasping strategies vary for each shape and it is assumed that the system has the knowledge about the shape prior to grasping from, for example, vision system. The type of grasps generated on objects of known shapes can easily be generated by a grasp planning system.

For shape independent approach, no knowledge of the object except the position of its center of mass with respect to the hand. Since the knowledge of the object shape is assumed unknown, there will be larger variation in the contact space and therefore more uncertainty in learning process. Therefore, the training data for this approach has been generated by sampling the grasps on a unit sphere with the origin in the object center. Example grasps are shown in Figure 2.

For shape specific approach, simulated datasets D_{S_1}, D_{S_2}, D_{S_3} are generated on a cylindric object and a bottle. Here, two types of grasps have been applied: a side and a top grasp. D_{S_1} and D_{S_2} include side grasps (for both objects) and D_{S_3} includes top grasps (for the bottle). Simulated datasets D_{S_4}, D_{S_5} are generated on a cylinder and a bottle by applying approach vectors sampled from a sphere around the object and including more than one preshape.

For labeling of the simulated grasp sequences we use grasp quality measure based on the radius of the largest enclosing

ball in the unit grasp wrench space (GWS) constructed as proposed in [18]. Two convex hulls, W_f and W_τ are calculated to separate wrench space with respect to forces and torques. Stable grasps are defined as those for which both quality values are within a threshold which has been set experimentally. The threshold for force takes the weight of the object into account by $x(m.g) \in W_f, x = 1.7$ so that the grasp remains stable even in case of additional forces.

The main purpose of the real world experiments is to demonstrate that the idea of grasp stability recognition is applicable in real-world scenarios. Thus, the experiments aim to serve as a proof-of-concept rather than assessing the exact performance rates in different use cases. We believe that performing real world experiments is important in order to validate the theoretical formalization and modelling.

For the real experiments, we have generated training data according to the shape specific strategy: the object shapes are assumed known and side and top grasps are applied on them. The objects are placed such that they are initially not well centered with respect to the hand to investigate the capability of the learning system to cope with potential uncertainties in the objects' pose. An example real grasp execution is shown in Figure 3.

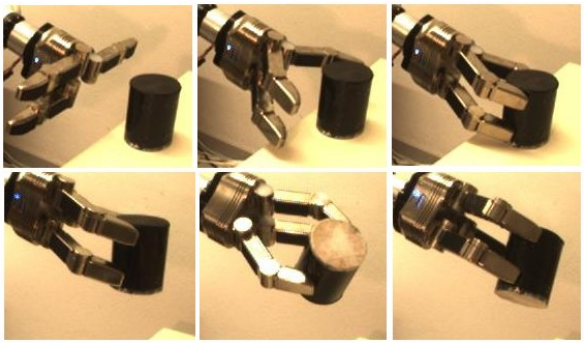


Fig. 3. A few examples from the execution of real experiments.

To generate the stable/unstable label for a grasping sequence, an object is lifted and rotated $[-120°, +120°]$ around the approach direction after a grasp has been applied to it. The grasps where the object is dropped or moved in the hand were labeled as unstable.

Training sequences $D_{R_1^2}$, $D_{R_2^2}$, $D_{R_3^2}$ are obtained by a parallel 2-fingered gripper with a deformable box and a deformable bottle shown in Figure 4. $D_{R_3^2}$ represents top grasps while the other two are side grasps. The rest of real

Fig. 4. Objects from the real datasets denoted by ($D_{R_1^2}$), ($D_{R_2^2}$, $D_{R_3^2}$), ($D_{R_1^3}$, $D_{R_4^3}$), ($D_{R_5^3}$), ($D_{R_2^3}$, $D_{R_3^3}$), ($D_{R_6^3}$) in the text.

data ($D_{R_1^3}$-$D_{R_6^3}$) are made on more rigid objects. $D_{R_1^3}$, $D_{R_2^3}$, $D_{R_3^3}$ are from the three fingered SDH and include contacts only on distal sensors: $D_{R_1^3}$ represents side grasps of a cylinder, $D_{R_2^3}$ side grasps of a bottle and $D_{R_3^3}$ top grasps of a bottle. $D_{R_4^3}$, $D_{R_5^3}$, $D_{R_6^3}$ are also side grasps for the same three-fingered hand but measurements from all six sensors are included.

VI. Experimental results

As mentioned, two HMMs, one for stable and another for unstable were trained with the stopping criteria being the convergence threshold 10^{-4} in maximum 10 iteration. Both ergodic and left-to-right HMMs were evaluated independently with different structure parameters. The range of 2-6 for the number of states and 2-5 for the number of components in a mixture were evaluated. Diagonal covariance matrix structure was chosen. By evaluating multiple temporal models we aim at understanding whether the temporal sequence plays part in the understanding of the grasp stability, or if only the final observation is sufficient.

Experiments were performed both on simulated and real data similarly. For simulated data 80% of the samples were used for training and 20% for testing. For the real data 10-fold cross validation was used to evaluate the performance. The number of stable and unstable samples are equal in each data set and the total number of samples are given in the Table I.

TABLE I
Number of Samples in Datasets

Data sets	Object	Grasp type	Number of samples
D_{S_1}	cylinder	side, 3-fingered	6400
D_{S_2}	bottle	side, 3-fingered	4906
D_{S_3}	bottle	top, 3-fingered	4446
D_{S_4}	cylinder	general, 3-fingered	6240
D_{S_5}	bottle	general, 3-fingered	2564
$D_{R_1^2}$	box	side, 2-fingered	148
$D_{R_2^2}$	bottle	side, 2-fingered	148
$D_{R_3^2}$	bottle	top, 2-fingered	100
$D_{R_1^3}$	cylinder	side, 3-fingered	140
$D_{R_2^3}$	bottle	side, 3-fingered	100
$D_{R_3^3}$	bottle	top, 3-fingered	50
$D_{R_4^3}$	cylinder	side, 3-fingered	60
$D_{R_5^3}$	cylinder	side, 3-fingered	60
$D_{R_6^3}$	bottle	side, 3-fingered	120

Table II presents the recognition rates on simulated data for the ergodic and left-to-right HMMs with the corresponding best parameter values. Ergodic and left-to-right HMMs have comparable results.

To illustrate the difference on performance for different objects, the distributions of logarithms of likelihood ratios are presented for two objects for the same type, ergodic HMM, in Figures 6 and 8. Let L_s be the log likelihood of the stable HMM model and L_u be the log likelihood of the unstable HMM model, then $r = L_s - L_u$ shows the log of the likelihood ratio. Figures 6, 8 show the histograms

TABLE II

RESULTS ON SIMULATED DATA

	D_{S_1}	D_{S_2}	D_{S_3}	D_{S_4}	D_{S_5}
$Rates_{ERG}$	**0.75**	**0.60**	**0.61**	**0.63**	**0.61**
$StableStates_{ERG}$	5	6	5	6	3
$StableComponents_{ERG}$	4	4	4	4	3
$UnstableStates_{ERG}$	4	5	6	5	2
$UnstableComponents_{ERG}$	4	3	3	5	4
$Rates_{LR}$	**0.75**	**0.60**	**0.61**	**0.65**	**0.62**
$StableStates_{LR}$	6	2	5	6	5
$StableComponents_{LR}$	4	5	4	2	2
$UnstableStates_{LR}$	4	4	5	3	4
$UnstableComponents_{LR}$	5	2	4	3	4
GMM	**0.76**	**0.59**	**0.59**	**0.57**	**0.60**
$GMMcomponents$	3	4	3	4	3

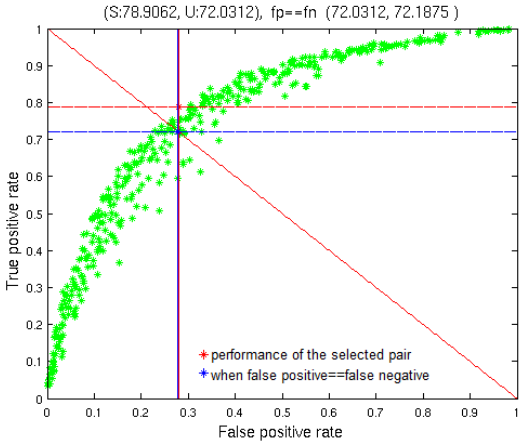


Fig. 5. The ROC for Cylinder side grasps.

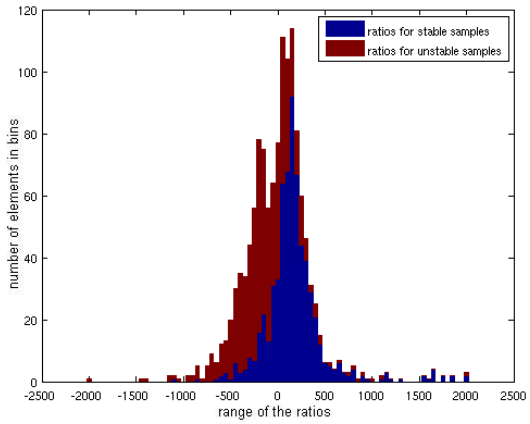


Fig. 6. The distribution of log-likelihood ratios for Cylinder side grasps.

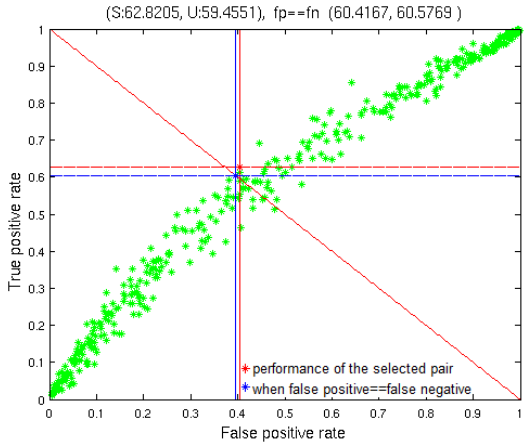


Fig. 7. The ROC for Bottle spherical grasps.

of these ratios (r) for stable and unstable samples. Blue bars show the difference for stable samples and red bars are for unstable samples. Figure 6 shows the distributions for the cylinder side grasps, for which the performance was relatively good, while in Figure 8 the distributions are given for the bottle grasps with spherical approach directions, for which the stability was more difficult to recognize. It is evident in the figures that the stable and unstable grasps differ reasonably.

Figures 5 and 7 with receiver operating characteristic (ROC) curves show how the HMM model parameters are chosen after training with different parameters. Each point in the figures indicate the performance of a trained HMM pair and the red cross indicates the performance of the selected HMM pair. Different HMM models were trained with different number of mixture components and states and finally the best HMM pair was chosen based on the maximum recognition rates for stable and unstable grasps. The blue lines crosses where the recognition performance gives equal number of false positives/negatives and the chosen HMM models give a performance around this point which is the best possible one among the trained models.

From Table III and Table IV, it is evident that the classification rates are reasonable for 2-fingered and 3-fingered grasps with real robots. Table V shows the performance of the HMM system for predicting the stability of the final grasp using the first half of sequences of the sensor readings. The HMMs were trained and tested with the first half of the training sequences.

As shown, the HMM results for the simulated data is similar to the one-shot approach. For the real data, one-shot and HMM results differ, which may mean that the process from the beginning to the end of the sequence has additional information that makes the HMM classification rate higher. We can note that the real data include readings from six tactile sensors while the simulated data include the readings from only three. Therefore, the contacts on the proximal sensors for the real experiments may hold additional information to reason about the stability.

Given the results, it is evident that the idea of using the tactile feedback to evaluate the stability of a grasp is applicable also in a real world scenario.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed the use of tactile sensing for estimating grasp stability using learning from training data. The experimental results show that tactile measurements

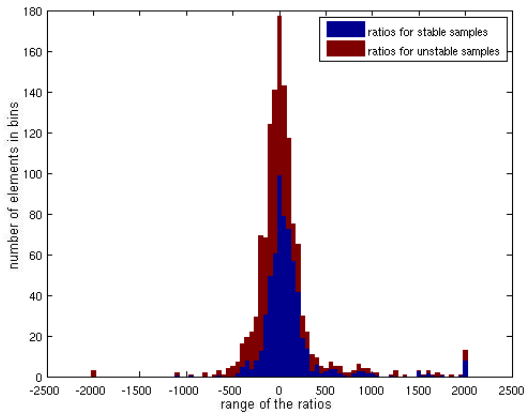


Fig. 8. The distribution of log-likelihood ratios for Bottle spherical grasps.

TABLE III

RESULTS ON REAL DATA WITH A 2 FINGERED GRIPPER

	$D_{R_1^2}$	$D_{R_2^2}$	$D_{R_3^2}$
$Rates_{ERG}$	**0.84**	**0.70**	**0.81**
$S.States_{ERG}$	2.5	3	3.1
$S.Components_{ERG}$	3.6	2.8	3.3
$U.States_{ERG}$	3.2	3.4	2.9
$U.Components_{ERG}$	3	3.3	3.4
$Rates_{LR}$	**0.85**	**0.71**	**0.74**
$S.States_{LR}$	3	2.7	2.8
$S.Components_{LR}$	4.1	2.5	2.8
$U.States_{LR}$	2.2	3.9	4
$U.Components_{LR}$	3.4	4.1	4

allow relatively good recognition of grasp stability, and that the ideas studied in simulation are also applicable in real robot systems. The aim of the paper was not a perfect discrimination between successful and unsuccessful grasps but rather a measure of certainty of grasp stability. This also means that the system may reject some stable grasps while having fewer unstable grasps classified as stable ones. We showed how a one-shot classifier and an HMM classifier perform with different datasets. Experiments showed that using sequential data to evaluate grasp stability appears to be beneficial during dynamic grasp execution.

Future work will be to first perform a more extensive evaluation of the method on more objects with more samples and also include all the sensors in simulation. We also plan to investigate the proposed idea on completely unknown objects by using data that includes multiple objects and then extend the methodology to evaluate part-based grasps.

REFERENCES

[1] J. Tegin, J. Wikander, S. Ekvall, D. Kragic, and B. Iliev, "Demonstration based learning and control for automatic grasping," in *International Conference on Advanced Robotics*, 2007.

[2] D. Kragic and H. I. Christensen, "Cue integration for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 17(1), pp. 18–27, 2001.

[3] V. Kyrki, D. Kragic, and H. I. Christensen, "New shortest-path approaches to visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 349–354.

[4] "Tactile database," http://www.nada.kth.se/∼yaseminb/.

[5] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer, 2008.

TABLE IV

RESULTS ON REAL DATA WITH THE SDH HAND

	$D_{R_1^3}$	$D_{R_2^3}$	$D_{R_3^3}$	$D_{R_4^3}$	$D_{R_5^3}$	$D_{R_6^3}$
$Rates_{ERG}$	**0.98**	**0.99**	**0.97**	**0.90**	**0.97**	**0.91**
$S.States_{ERG}$	2.7	2.2	2.2	2.1	2.7	2.3
$S.Components_{ERG}$	2.7	2.8	2	3.7	2.8	3.8
$U.States_{ERG}$	2.1	2.6	2.1	2.4	2.2	2.5
$U.Components_{ERG}$	2.6	2.9	2.6	3	2.5	3.8
$Rates_{LR}$	**0.99**	**0.98**	**0.96**	**0.93**	**0.98**	**0.93**
$S.States_{LR}$	2.6	2	2.4	2.8	3.2	3.8
$S.Components_{LR}$	3.1	2.4	2.1	3.5	2.9	3.6
$U.States_{LR}$	2	2.1	2.1	2.6	2.5	3.4
$U.Components_{LR}$	2.5	2.9	2.2	2.4	2.5	3.5

TABLE V

RESULTS USING SUBSEQUENCES TO PREDICT THE STABILITY OF THE FINAL GRASP

	D_{S_1}	D_{S_4}	D_{S_5}	$D_{R_5^3}$	$D_{R_6^3}$
$Rates_{LR}$	**0.68**	**0.55**	**0.54**	**0.90**	**0.88**
$S.States_{LR}$	6	3	4	2.7	3.6
$S.Components_{LR}$	5	5	3	3.3	3.2
$U.States_{LR}$	6	4	4	2.3	2.2
$U.Components_{LR}$	5	4	2	2.3	3.8
$GMMrates$	**0.68**	**0.57**	**0.55**	**0.79**	**0.78**

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

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

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

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

[10] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online grasp planning," *IEEE International Conference on Robotics and Automation, ICRA'01*, pp. 2460–2465, 2001.

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

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

[13] J. Romero, H. Kjellstrom, and D. Kragic, "Markerless human-to-robot grasp mapping based on a single view," in *International Conference on Advanced Robotics*, 2009.

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

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

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

[17] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.

[18] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE Int. Conf, on Robotics and Automation*, 1992, pp. 2290–2295.

Using context to identify novelty during simple manipulation of rigid objects

Sebastian Zurek, Marek Kopicki, Rustam Stolkin, and Jeremy Wyatt
School of Computer Science
University of Birmingham, UK

July 29, 2010

Abstract

We adapt a model of human sensorimotor learning and control to the robotic domain. The modular motor learning theory of Wolpert and Kawato makes use of a set of motor controllers, in which each controller is suitable for one or a few contexts. Here a context is understood to be a configuration of the environment, such as object weight or shape. We apply this idea of context to predict the motion of a rigid object manipulated by a robotic finger. Given a trained set of predictors, Bayesian model selection is used to infer the context during a manipulation experiment. To detect novel contexts, a "novelty" predictor competes with the trained predictors in the model selection process. Preliminary results from an experimental trial, in which an object is pushed by a robotic finger, demonstrate how the estimate of context varies with time.

1 Introduction

Predicting the motion of a rigid object undergoing manipulation by a robotic device is challenging task, but key to devising robotic control and planning systems. In this report we focus on the use of context to model object motion caused by robotic manipulation, and on the detection of novelty from the resultant object behaviour. In order to predict the behaviour of a rigid object subjected to a simple manipulation, such as a push, we accept that uncertainty arises in several aspects of the prediction system's knowledge about the object, namely:

1. the precise trajectory realized by the object during manipulation
2. the object's identity (which determines intrinsic characteristics of the object, such as shape, size, weight, mass distribution, and surface frictional properties)
3. whether the object is novel or already known to the prediction system

Here we will restrict attention to just the third aspect, and consider a method to detect novel objects by observing their motion under manipulation.

In section 2 we explore the notion of context and how it can be utilized to predict object behaviour. Section 3 briefly sketches a probabilistic representation of rigid object motion that is detailed elsewhere [1]. In section 4 we present a Bayesian approach to context-based prediction of object motion and outline how to perform context estimation and novelty detection. Some preliminary results from robotic experiments are reported in section 5.

2 Utilizing context

To formulate a model that utilizes context, first we attempt to be more precise about the notion of context. We can list several aspects of context:

cardinality of domain: discrete or continuous (or hybrid)

visibility: (directly) observed or hidden

temporal: constant or time-varying

mixing: are different contexts disjoint or can they occur in blends?

shared structure: do contexts share some latent structure?

There is a relationship between the cardinality and mixing aspects, in that discrete contexts are naturally disjoint, whereas continuous-valued contexts can admit blends. However, it is less clear whether it is appropriate to form a blend from discrete contexts. More generally, a context can be some configuration of the environment relevant to the task of interest.

Probabilistic models of object motion typically have a high-dimensional input domain, which leads to slow learning and poor generalization, unless adequate regularization is used. These models can be augmented with context parameters that act to partition the input domain and hence reduce its effective dimensionality.

Previous work by S. Vijayakumar and colleagues has explored these issues for the task of adaptive motor control in robotics. For example, the following two cases were investigated by Petkos and Vijayakumar [2]:

1. multiple (inverse) models with discrete, hidden, time-varying, disjoint contexts.

2. a single model with a set of continuous, hidden, time-varying context variables, in which contexts could be blended, used to control a robotic arm loaded with an unknown mass.

In [2] it was argued that the multiple model paradigm (case 1) had difficulty dealing with novel contexts, and that there was no obvious way to generalize between contexts (by e.g. blending). Instead a single model (case 2) was proposed that was augmented with a set of hidden context variables. However,

learning and context estimation were only possible due to the special structure of the model. In general, this approach would require a very large set of training samples to succeed.

Thus we return to consider discrete contexts, with a mechanism to detect novelty and to create new models. This approach still has limitations in those situations where context is naturally continuous, such as the weight of an object (and indeed case 2 above). An attempt could be made to cover a range of weights by a set of discrete contexts, and assume a linear blending rule to interpolate for an arbitrary weight in the range.

To add context to a predictive model, the following computational tasks need to be addressed:

1. online context estimation
2. data allocation for training models
3. when and how to create a new model, i.e. novelty detection

3 Representation of simple manipulations of a rigid object by a robotic finger

The predicted pose B_{t+1} of a manipulated object is represented as a product of probability distributions P_G and P_L:

$$P(B_{t+1}|X_t) = \frac{1}{Z_1} P_G(B_{t+1}|X_t^{(G)}) P_L(B_{t+1}|X_t^{(L)}) \tag{1}$$

where the X_t are conditioning variables known at time t, and Z_1 is a normalization constant. In [1], the probability distributions P_G and P_L are called the "global expert" and "local expert" respectively. The conditioning variables X_t are functions of the robot finger trajectory $A_{0:T}$ (assumed deterministic and known in advance), and the current object pose B_t.

The distributions P_G and P_L are represented as mixtures of gaussians (MoG) and are learned using a kernel density estimation procedure. Further details are given in [1].

4 Prediction of object motion and context estimation

We now detail a Bayesian model selection approach [3] to implementing a context-based predictor for the motion of an object during robotic manipulation.

Consider a context predictor with K models M_1 to M_K, corresponding to K discrete contexts. Let $B_{0:t}$ denote the set of observations (i.e. trajectory of pushed object) up to time t. Then the posterior probability of model M_k given the observed data is

$$P(M_k|B_{0:t}) = \frac{P(B_{0:t}|M_k)P(M_k)}{P(B_{0:t})}. \tag{2}$$

If we take all models to be equally likely at the start i.e. $P(M_k) = 1/K$ we have

$$P(M_k|B_{0:t}) = \frac{P(B_{0:t}|M_k)}{\sum_{k=1}^{K} P(B_{0:t}|M_k)}. \tag{3}$$

To deal with novel contexts we introduce an extra model M_0 that dominates when all the other models assign a low probability to the observed trajectory. Equation 2 is still valid, but we will rewrite the priors $P(M_k)$ as

$$P(M_k) \rightarrow \begin{cases} (1-\epsilon)\widetilde{P}(M_k) = (1-\epsilon)/K, & k \neq 0 \\ \epsilon, & k = 0 \end{cases} \tag{4}$$

so that $\sum_{k=0}^{K} P(M_k) = 1$. Here ϵ is the (small) prior probability of a novel context given that no observations have been made. It remains to compute the likelihoods $P(B_{0:t}|M_k)$. For $k = 0$, the likelihood is assumed to be a constant over the parameter space spanned by the object pose B_t, which is a region of $\mathbb{R}^6$ corresponding to physically feasible values of B_t (see [1]). The next subsections detail the calculation for $k \neq 0$.

4.1 Calculation of likelihood $P(B_{0:t}|M_k)$

$P(B_{0:T}|M_k)$ is the probability of seeing the trajectory $B_{0:T}$ assuming model M_k obtains. In general it can be written as a product

$$P(B_{0:T}|M_k) = \prod_{t=1}^{T} P(B_t|B_{0:t-1}, M_k). \tag{5}$$

The prediction model in [1] makes use of a quasi-static approximation so that the prediction of the object pose at time t depends only on state at time $t-1$. Hence $P(B_t|B_{0:t-1}, M_k) = P(B_t|B_{t-1}, M_k)$ is Markovian and we can simplify equation 5 to

$$P(B_{0:T}|M_k) = \prod_{t=1}^{T} P(B_t|B_{t-1}, M_k) \tag{6}$$

which can also be expressed as a recursion in $L_T \triangleq P(B_{0:T}|M_k)$

$$L_T = L_{T-1} P(B_T|B_{T-1}, M_k). \tag{7}$$

So to calculate the likelihood we can update a variable L_T by multiplying by the probability of seeing the observed transition $B_{T-1} \rightarrow B_T$ according to model M_k. Unfortunately the desired quantity $P(B_T|B_{T-1}, M_k)$ is not readily available from the prediction scheme used in [1], since a prediction requires only an unnormalized score $\gamma(B_T|\cdot)$ where

$$P(B_T|\cdot) = \frac{\gamma(B_T|\cdot)}{Z} \tag{8}$$

and Z is a normalization constant (conditioning variables have been suppressed). In [1], $\gamma(B_T|\cdot)$ is the unnormalized product of probabilities from several "experts" – hence $P(B_T|\cdot)$ is a product density.

4.2 Calculation of normalization constant for the product density

Thus to compute $P(B_T|B_{T-1}, M_k)$ we are required to normalize the density arising from the product of experts. Each expert P_G and P_L is modelled as a mixture of gaussians (MoG) with respect to some predicted variable. If all experts predicted the same variable (e.g. change in pose of pushed object $T(B_t, B_{t-1})$) then the product density $P(B_T|B_{T-1}, M_k)$ would also be a MoG, albeit with N^c terms, where N is the number of gaussians in a mixture and c is the number of experts. However, the experts differ in the variable that is predicted, so that Jacobians have to be introduced in order to compute the normalization integral

$$Z = \int \gamma(B|\cdot)dB. \tag{9}$$

This leads to further difficulties, so an approximate transform is used that maps the predicted variable of all experts to the predicted object pose B_{t+1}. Still the resulting normalization integral cannot be computed analytically, so a Monte Carlo scheme based on importance sampling is employed.

Let $q(B)$, known as the importance or proposal distribution, be a probability density from which samples can be drawn easily. Define $\phi(B)$ as

$$\phi(B) \triangleq \frac{\gamma(B|\cdot)}{q(B)}. \tag{10}$$

where conditioning variables have been suppressed. Then draw N samples $\widetilde{B_i}$ from $q(B)$. An approximation for Z is given by

$$Z = \mathbb{E}^{\,q}[\phi(B)] \approx \frac{1}{N}\sum_{i=1}^{N}\phi(\widetilde{B_i}). \tag{11}$$

The quality of the approximation depends on how closely $q(B)$ resembles $P(B_T|\cdot)$. One suggestion is to use the "global" expert density P_G as the importance density. This density is a MoG, which is straightforward to normalize and sample.

To derive the first equality in equation 11, we recall (subject to some technical conditions) that for an arbitrary function $\phi(B)$

$$\mathbb{E}^{\,q}[\phi(B)] = \mathbb{E}^{\,P}\left[\phi(B)\frac{q(B)}{P(B)}\right], \tag{12}$$

where $q(B)$ and $P(B)$ are two probability distributions. Substituting for $\phi(B)$ using equations 10 and 8, immediately we obtain

$$\mathbb{E}^{\,P}\left[\frac{\gamma(B)}{q(B)}\frac{q(B)}{P(B)}\right] = \mathbb{E}^{\,P}[Z] = Z, \tag{13}$$

where the last equality holds since Z is a constant.

4.3 Context estimation

An estimate of the context $\hat{k}$ at time t is given by

$$\hat{k} = \underset{k}{\operatorname{argmax}}\, P(M_k|B_{0:t}) \tag{14}$$

A novel context is detected if $\hat{k} = 0$, which leads to the creation of a new model, and hence increases the total number of models $K \rightarrow K + 1$. Note that in contrast to other work [4, 2], we do not make use of a transition model, such as a HMM, to model context dynamics.

4.4 Data allocation during learning

During learning each model M_k has to assess and weight training data that is obtained from observing and interacting with objects. In general, this data allocation task is a challenging problem if contexts can change in early phases of learning, since models will typically have poor accuracy which will prevent correct context identification.

However, our experiments were structured as a set of short trials, where for each trial the robot interacted with just a single rigid object. Since the context is taken to be a function of the object's properties, it will be assumed that the context remains fixed over the course of a single trial. Thus the initial portion of a trial can be devoted to determining the context, which once established is used to allocate data over the remainder of the trial.

4.5 Prediction

In [1], prediction is based on the MAP estimate of the product density $P(B_{t+1}|\cdot)$, where B_{t+1} is the object pose predicted for the next time step. With multiple models, this prediction scheme can be augmented to utilize context. One option is to use a winner-take-all mechanism, i.e. take

$$P(B_{t+1}|X_t, M_{\hat{k}}) \tag{15}$$

for the MAP optimization, where $\hat{k}$ is given by equation 14. Alternatively for some contexts, it may be appropriate to use Bayesian model averaging in which we form a blend by marginalizing out M_k

$$P(B_{t+1}|\cdot) = \sum_{k=0}^{K} P(B_{t+1}|X_t, M_k)P(M_k|B_{0:t}) \tag{16}$$

More formally one can use a decision theoretic framework and introduce a utility function, in which case the above two variants can be derived for suitable choices of utility function.

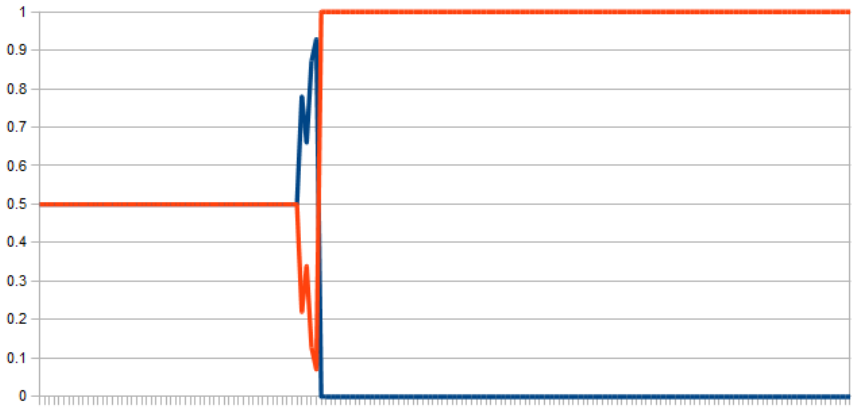


Figure 1: Posterior probability of each model $P(M_k|B_{0:t})$ as a function of time t during a single trial. Here the total number of models $K = 2$, with no novelty model M_0. The robot finger first made contact with the object several seconds into the trial.

5 Experimental Results

To demonstrate the context predictor, we trained two models $k = 1, 2$ on $n = 150$ pushing trials. Model M_1 was trained on a smooth metal polyflap, whereas M_2 was trained on a polyflap coated with coarse-grade sandpaper (and hence altered sliding friction).

Figure 1 shows the posterior probability of each model $P(M_k|B_{0:t})$ as a function of time t for a single test trial. For about the first third of the trial the robot finger is not in contact with the polyflap, so the posteriors are equiprobable. On making contact with the polyflap, the posterior oscillates for a few seconds before settling on the correct k.

The sequence of object poses was obtained by visual tracking [5].

6 Conclusion

We have described a context-based probabilistic method that predicts the resultant motion of a rigid body when acted upon by a robotic manipulator. Preliminary robotic experiments have provided initial validation of the model, but further issues need to be addressed.

During the course of a pushing trial, the estimated context was seen to fluctuate wildly. In part this is due to observation noise, which has been ignored in the modelling thus far. The time interval between observations was rather short ($\sim 0.1s$) which would exacerbate the difficulty of discriminating between contexts.

This work has also finessed the data allocation problem (§4.4). However, it is not unreasonable to train the predictors at least initially with some labelled data, as in curriculum learning.

We intend to conduct further work to assess the robustness and scalability of the prediction scheme, especially the interaction between context estimation and data allocation.

Acknowledgement

The work described in this report was funded by the European Commission's Seventh Framework Programme, contract no. ICT-215181-CogX.

References

[1] M. Kopicki, R. Stolkin, S. Zurek, and J. Wyatt, "Learning to predict how rigid objects behave under simple manipulation," tech. rep., School of Computer Science, University of Birmingham, 2010.

[2] G. Petkos and S. Vijayakumar, "Context estimation and learning control through latent variable extraction: From discrete to continuous contexts," in *Proc. of the International Conference on Robotics and Automation (ICRA'07)*, 2007.

[3] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, Jun 1995.

[4] M. Haruno, D. M. Wolpert, and M. Kawato, "MOSAIC model for sensorimotor learning and control," *Neural Computation*, vol. 13, p. 2201, 2001.

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