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

Deliverable Number:	D12
Deliverable Title :	Occulomotor vision based system for object and action learning
Type (Internal, Restricted, Public):	PU
Authors	D. Kragic, D. Song, J. Bohg, T. Asfour, R. Dillmann, M. Do, P. Pastor, T. Feix
Contributing Partners	KTH, UniKarl, Otto-Bock

Contractual Date of Delivery to the EC:	28-02-2010
Actual Date of Delivery to the EC:	28-02-2010

Contents

Chapter 1

Executive summary

Deliverable D12 presents the second year developments within WP2 - "Representations and Ontology for learning and Abstraction of Grasping". According to the Technical Annex, deliverable D12 presents the activities in the context of Tasks 2.1-2.3:

- [**Task 2.1**] - Definition of the ontology: definition of sensory-motor control for action and object-action learning
- [**Task 2.2**] - Vocabulary of human and robot actions/interactions
- [**Task 2.3**] - Evaluation of representation: Evolving ontology through modeling of the perception-action cycle

The work in this deliverable relates to the following second year Milestones:

- [**Milestone 4**] - Analysis of action-specific visuo-spatial processing vocabulary of human actions/interactions for perception of task relations and affordances
- [**Milestone 6**] - Integration and evaluation of human hand and body tracking on active robot heads, demonstration of a grasping cycle on the experimental platforms

The progress in WP2 is presented in the below summarized scientific publications, attached to this deliverable.

- In Attachment A we present work on active vision based sensing and control required by Task 2.1. The work on finding, attending, recognizing and manipulating objects in domestic environments is studied. We present a stereo based vision system framework where aspects of Top-down and Bottom-up attention as well as foveated attention are put into focus and demonstrate how the system can be utilized for robotic object grasping. The system is a necessary building block for Task 2.3.
- In Attachment B we continue the work in Attachment A and present an active vision system for segmentation of visual scenes based on integration of several cues. The system serves as a visual front end for generation of object hypotheses for new, previously unseen objects in natural scenes necessary for the work in WP 4 and WP 5. The system combines a set of foveal and peripheral cameras where, through a stereo based fixation process, object hypotheses are generated. In addition to considering the segmentation process in 3D, the main contribution of the paper is integration of different cues in a temporal framework and improvement of initial hypotheses over time.
- In Attachment C we present work on studying human grasping actions. This work related to the scientific questions studied in WP 1 and relates to Task 2.2. Understanding the spatial dimensionality and temporal context of actions can be used to i) provide representations for programming grasping actions in robots, and ii) use these for designing new robotic and prosthetic hands.

  Natural human hand motion is highly non-linear and of high dimensionality. However, for specific activities such as handling and grasping of objects, the commonly observed hand motions lie on a lower-dimensional non-linear manifold in hand posture space. This is also true for human motion in general. Although full body human motion is well studied within Computer Vision and Biomechanics, there is very little work on the analysis of hand motion. We use Gaussian Process Latent Variable Models (GPLVMs) to model

the lower dimensional manifold of human hand motions, during object grasping in particular. We show how the technique can be used to embed high-dimensional grasping actions in a lower-dimensional space suitable for modeling, recognition and mapping.

- In Attachment D we present the work related to Task 2.2 and Task 2.3. The work studies the learning of task-relevant features that allow grasp generation in a goal-directed manner. We show how an object representation and a grasp generated on it can be integrated with the task requirements. The scientific problems tackled are (i) identification and modeling of such task constraints, and (ii) integration between a semantically expressed goal of a task and quantitative constraint functions defined in the continuous object-action domains. We first define constraint functions given a set of object and action attributes, and then model the relationships between object, action, constraint features and the task using Bayesian networks. The probabilistic framework deals with uncertainty, combines a-priori knowledge with observed data, and allows inference on target attributes given only partial observations. We present a system designed to structure data generation and constraint learning processes that is applicable to new tasks, embodiments and sensory data. The application of the task constraint model is demonstrated in a goal-directed imitation experiment.

- In Attachment E we study the problem of hand-eye coordination and, more specifically, tool-eye recalibration of humanoid robots, related to Task 2.2. Inspired by results from neuroscience, a novel method to learn the forward kinematics model as part of the body schema of humanoid robots is presented. By making extensive use of techniques borrowed from the field of computer-aided geometry, the proposed *Kinematic Bézier Maps (KB-Maps)* permit reducing this complex problem to a linearly-solvable, although high-dimensional, one. Therefore, in the absence of noise, an *exact* kinematic model is obtained. This leads to rapid learning which, unlike in other approaches, is combined with robustness to sensor noise and good extrapolation capabilities. These promising theoretical advantages have been validated through simulation, and the applicability of the method to real hardware has been demonstrated through experiments on the humanoid robot ARMAR-IIIa.

Appendix A

Attached Papers

[A] **An active vision system for detecting, fixating and manipulating objects in real world**, International Journal of Robotics Research, 2009.

[B] **Active 3D scene segmentation and detection of unknown objects**, M. Bjorkman, D. Kragic, IEEE International Conference on Robotics and Automation, 2010.

[C] **Spatio-temporal modeling of grasping actions** J. Romero, T. Feix, H. Kjellstrom, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. (submitted)

[D] **Learning Task Constraints for Robot Grasping using Graphical Models**, D. Song, K. Huebner and D. Kragic, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. (submitted)

[E] **Rapid Learning of Humanoid Body Schemas with Kinematic Bezier Maps**, S. Ulbrich, V. Ruiz de Angulo, T. Asfour, C. Torras, R. Dillmann, Humanoids 2009.

An Active Vision System for Detecting, Fixating and Manipulating Objects in Real World

B. Rasolzadeh, M. Björkman, K. Huebner and D. Kragic

August 24, 2009

Abstract

The ability to autonomously acquire new knowledge through interaction with the environment is an important research topic in the field of robotics. The knowledge can be acquired only if suitable perception-action capabilities are present: a robotic system has to be able to detect, attend to and manipulate objects in its surrounding. In this paper, we present the results of our longterm work in the area of vision based sensing and control. The work on finding, attending, recognizing and manipulating objects in domestic environments is studied. We present a stereo based vision system framework where aspects of Top-down and Bottom-up attention as well as foveated attention are put into focus and demonstrate how the system can be utilized for robotic object grasping.

1 Introduction

Humans use visual feedback extensively to plan and execute actions. However, this is not a well-defined one way stream: how we plan and execute actions depends on what we already know about the environment we operate in (context), what we are about to do (task), and what we think our actions will result in (goal). A significant amount of human visual processing is not accessible to consciousness - we do not *experience* using optical flow to control our posture, (Sloman, 2001). By not completely understanding the complex nature of human visual system, what are the ways to model similar capabilities into robots?

Visual attention plays an important role when we interact with the environment, allowing us to deal with the complexity of everyday scenes. The requirements on artificial 'seeing' systems are highly dependent on the task and have historically been developed with this in mind. To deal with the complexity of the environment, prior task and context information have commonly been integrated with low level processing structures, the former being denoted as Top-down and latter Bottom-up principle.

In our research, tasks such as "Robot, bring me my cup" or "Robot, pick up this" are studied. Depending on the task or context information, different execution strategies task may be chosen. The first task is well defined in that

manner that the robot already has the internal representation of the object - the *identity* of the object is known. For the second task, the spoken command is commonly followed by a pointing gesture - here, the robot does not know the *identity* of the object, but it can rather easy extract its *location*. A different set of underlying visual strategies are required for each of these scenarios being the most representative examples for robotic fetch-and-carry tasks. We have worked with different aspects of the above for the past several years, (Björkman and Eklundh, 2002; Björkman and Eklundh, 2006; Björkman and Kragic, 2004; Ekvall and Kragic, 2005; Ekvall *et al.*, 2007; Huebner *et al.*, 2008a; Kragic and Kyrki, 2006; Petersson *et al.*, 2002; Rasolzadeh *et al.*, 2006, 2007; Topp *et al.*, 2004). The work presented here continues our previous works, but the focus is put on the design and development of a vision system architecture for the purpose of solving more complex visual tasks. The overall goal of the system is to enable the understanding and modeling of space in various object manipulation scenarios. A schematic overview of the experimental platform is shown in Fig. 1. The different parts of this illustration are described in detail throughout this paper.

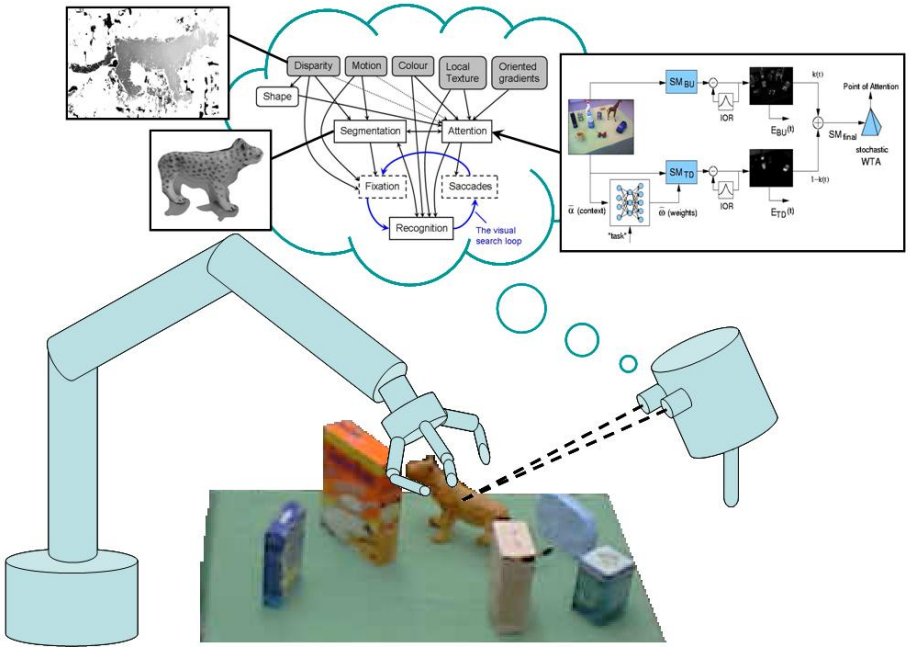


Figure 1: Illustration of the complete robotic platform that is the system described in this paper. See Fig. 2 and 4 for detailed illustrations, and Fig. 10 for an illustration of the actual setup.

The paper is organized as follows. In Section 2 system functionalities, considered tasks, system design and flow of information in the system are outlined.

This corresponds roughly to the diagram in the middle of Fig. 1. In Section 3, the details about the camera system and its calibration are given. Aspects of Bottom-up and Top-down attention are presented in Section 4 and foveated segmentation in Section 5. Section 6 describes how the visual system can be used to facilitate manipulation. Selected results of the experimental evaluation are presented in Section 7, where an evaluation of the attention-system and the recognition-system is first performed separately, followed by a find-and-remove-object task in Section 7.3. A discussion and summary finalizes the paper in Section 8.

2 Vision System Functionalities and Tasks

Similar to the human vision system, but unlike many systems from the computer vision community, robotic vision systems are embodied. Vision is used as a mean for the robot to interact with the world. The system perceives to act and acts to perceive. The vision system is not an isolated entity, but part of a more complex system. Thus, the system should be developed and evaluated as such. In fact, measuring the performance of system components in isolation can be misleading. The quality of a component depends on its ability to function within the rest of the system. Computational speed might sometimes be preferable to accuracy or vice-versa. As a designer, one has to take a step backwards and concentrate on the tasks the robotic system is supposed to perform and the world in which the system resides. What are the goals of the system? What can be expected from the world and what cannot be expected?

Examples of recent work taking the embodiment into account have been demonstrated in Björkman and Eklundh (2006); Ude *et al.* (2006). In these systems vision is embodied in a robotic system capable of visual search as well as simple object manipulation. The goal of the work presented here is to design a similar robotic system able to interact with the world through recognition and manipulation of objects. Objects can either be previously known or completely new to the system. Even if confusions do occur frequently, a human being is able to immediately divide the perceived world into different physical objects, seemingly without effort. The task is performed with such ease that the complexity of the operation is easily underestimated. For a robotic system to perform the same task, the visual percept has to be grouped into larger entities that have some properties in common, properties such as proximity and appearance. These perceptual entities might or might not correspond to unique physical objects in 3D space. It is not until the robot acts upon an entity, that the existence of a physical object can be verified. Without interaction the entity has no real meaning to the robot. We call these entities *things* to differentiate them from *objects* that are known to be physical objects, through interaction or other means. The idea of *things* and *objects* is the foundation of the project PACO-PLUS [1] and the recent work presented in Geib *et al.* (2006); Kraft *et al.* (2008); Wörgötter *et al.* (2009).

[1] www.paco-plus.org

For the visual system to be of use in robotic tasks, it needs the abilities to divide the world into *things*, represent these for later association and manipulation, and continuously update the representation as new data becomes available. A representation can either be short-lived and survive only during a short sequence of actions, or permanent, if interactions with the *thing* turn out to be meaningful. A meaningful action is an action that results in some change in the representation of the *thing*, such as a pushing action resulting in a change in position. From this stage on, the *thing* is considered an *object*.

The amount of perceptual data arriving through a visual system easily becomes overwhelming (Tsotsos, 1987). Since resources will always be limited in one way or the other, there is a need for a mechanism that highlights the most relevant information and suppresses stimuli that is of no use to the system. Instead of performing the same operations for all parts of the scene, resources should be spent where they are needed. Such a mechanism is called *visual attention*. Unfortunately, relevancy is not a static measure, but depends on the context, on the scene in which the robot acts and the tasks the robot is performing. Consequently, there is a need for the attentional system to adapt to context changes as further studied in Section 4. A static *thing* too large for the robot to manipulate might be irrelevant, while an independently moving *thing* of the same size can be relevant indeed, if it affects the robot in achieving its goals. Since sizes and distances are of such importance to a robotic system, a visual system should preferably consist of multiple cameras.

2.1 Flow of Visual Information

The visual system used in our study has a set of basic visual functionalities, the majority of which uses binocular cues, when such cues are available. The system is able to attend to and fixate on *things* in the scene. To facilitate object manipulation and provide an understanding of the world, there is support for figure-ground segmentation, recognition and pose estimation. All these processes work in parallel, but at different time frames, and share information through asynchronous connections. The flow of visual information through the system is summarized in Fig. 2. Information computed within the system is shown in rounded boxes. Squared boxes are visual processes that use this information. Grey boxes indicate information that is derived directly from incoming images. The camera control switches between two modes, fixation and saccades, as illustrated by the dashed boxes. The vision system generally works within the visual search loop that consists of a saccade to the current attentional foci, after which the system tries to fixate on that point, which in turn will yield more (3D) information for the recognition step. If the attended/fixated region is not the desired object we are searching for, the visual search loop continues.

The above-mentioned vision system has been implemented on the four-camera stereo head (Asfour *et al.*, 2006) shown in Fig. 10. The head consist of two foveal cameras for recognition and pose estimation, and two wide field cameras for attention. It has seven mechanical degrees of freedom; neck roll, pitch and yaw, head tilt and pan & tilt for each camera in relation to the neck. The

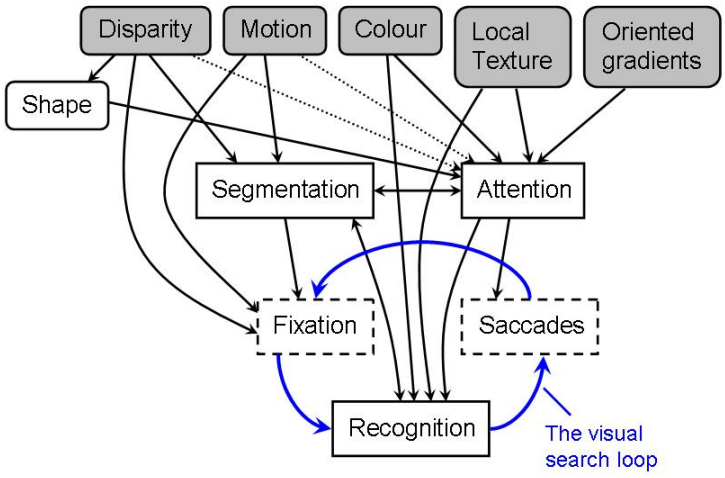


Figure 2: The flow of visual information.

attentional system keeps updating a list of scene regions that might be of interest to the rest of the system. The oculomotor system selects regions of interest from the list and directs the head so that a selected region can be fixated upon in the foveal views. Redirection is done through rapid gaze shifts (saccades). As a consequence, the camera system always strives towards fixating on some region in the scene. The fact that the system is always in fixation is exploited for continuous camera calibration and figure-ground segmentation.
Given the large focal lengths of the foveal cameras, the range of possible disparities can be very large, which complicates matching in stereo. With the left and right foveal cameras in fixation, we know that an object of interest can be found around zero disparity. This constrains the search range in disparity space, which simplifies stereo matching and segmentation.

2.2 Design Issues

We have chosen a design methodology that is biologically inspired, without the ambition to make our systems biologically plausible. Since computational and biological architectures are so fundamentally different, biologically plausibility comes at a cost. One critical difference is the relative costs of computation and communication of the estimated results. In biological systems, computations are done in neurons, with results communicated through thousands of synapses per neuron. This is much unlike computational systems in which the cost of communicating data, through read and write operations to memory, can be higher than that of computing the actual data. Unfortunately, computer vision tend to be particularly memory-heavy, especially operations that cover whole images. If one considers typical real-time computer vision systems, the cost of storage easily out-weight the cost of computation. Thus for a system to perform

at real-time speed, biological plausibility easily becomes a hindrance. Even if biological systems inspire the design process, our primary interest is that of robotic manipulation in domestic environments.

3 Camera System

For a robot to successfully react to sudden changes in the environment the attentional system ought to cover a significant part of the visual field. Recognition and vision-based navigation, however, place another constraint on the visual system, i.e. a high resolution. Unfortunately, these two requirements are hard to satisfy for a system based on a single stereo pair. A biological solution, exemplified by the human vision system, is a resolution that varies across the visual field, with the highest resolution near the optical centers. There are some biologically-inspired artificial systems (Kuniyoshi *et al.*, 1995; Sandini and Tagliasco, 1980) that use similar approaches. However, non-uniform image sampling leads to problems that make these systems less practical. Binocular cues can be beneficial for a large number of visual operations, from attention to manipulation, and with non-uniform sampling stereo matching becomes hard indeed. Furthermore, the reliance on specialized hardware makes them more expensive and less likely to be successfully reproduced. Another possible solution is the use of zoom lenses (Paulus *et al.*, 1999; Ye and Tsotsos, 1999). While the robot is exploring the environment the lenses are zoomed out in order to cover as much as possible of the scene. Once an object of interest has been located, the system zooms in onto the object to identify and possibly manipulate it. However, while the system is zoomed in it will be practically blind to whatever occurs around it.

Other than the obvious reasons of cost and weight, there is nothing conceptually preventing us from using more than just two cameras, which is the case in solutions based on either zoom-lenses or non-uniform sampling. Instead, one could use two sets of stereo pairs (Scassellati, 1998), a wide-field set for attention and a foveal one for recognition and manipulation. The most important disadvantage is that the calibration process becomes more complex. In order to relate visual contents from one camera to the next, the relative placement and orientation of cameras have to be known.

Sets of four cameras can be calibrated using the quadrifocal tensor (Hartley and Zisserman, 2000), or the trifocal tensor if sets of three are considered at a time. However, the use of these tensors assumes that image features can be successfully extracted and matched between all images considered. Depending on the camera configuration and observed scene, it may not at all be the case. For example, due to occlusions the visual fields of the two foveal images might not overlap. Furthermore, since the visual fields of the foveal cameras are so much narrower than those of the wide-field ones, only large scale foveal features can be matched to the wide-field views. The largest number of matchable features is found if only two images are considered at a time and the corresponding focal lengths are similar in scale. Thus for the system presented in this paper, we use

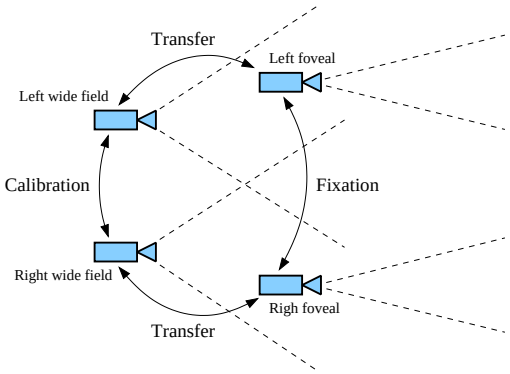


Figure 3: Two sets of cameras, a wide-field camera set for attention and a foveal one for recognition and manipulation, with external calibration performed between pairs.

pair-wise camera calibration as illustrated by the arrows in Fig. 3.

3.1 Wide-Field Calibration

Since external calibration is inherently more stable if visual fields are wide, we use the wide-field cameras as references for the foveal ones. This calibration is an on-going process, where previous estimates are exploited for feature matching in the subsequent frames, assuming a limited change in relative camera orientation from one frame to the next. The purpose of the calibration is two-fold. First, given a known baseline (the distance between the cameras) it provides a metric scale to objects observed in the scene. Second, calibration provides parameters necessary for rectification of stereo images, so that dense disparity maps can be computed, as it will be shown in Section 4.4.

In our study we related the wide-field cameras to each other using an iterative approach based on the optical flow model (Longuet-Higgins, 1980):

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} 1 & x \\ 0 & y \end{pmatrix} \begin{pmatrix} t_x \\ t_z \end{pmatrix} + \begin{pmatrix} 0 & 1+x^2 & -y \\ 1 & xy & x \end{pmatrix} \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \quad (1)$$

In an earlier study (Björkman and Eklundh, 2002), we have shown this model to more gracefully degrade in cases of image noise and poor feature distributions, than the more popular essential matrix model (Longuet-Higgins, 1981). The rotational (r_x, r_y, r_z) and translational (t_x, t_z) parameters are determined iteratively from matches of corner features. We use Harris' corner features (Harris and Stephens, 1988) for the purpose and apply random sampling (Fischler and Bolles, 1981) to reduce the influence from outliers in the dataset. Matching is done using normalized cross-correlation of 8×8 pixel image patches. As quality measure we use a left-to-right and right-to-left matching consistency check, rather than thresholding on matching scores. Once the above parameters are

known, the metric distance to the current fixation point is computed from the baseline and the vergence angle r_y. This distance is later used as a reference for distance and scale measurements, as well as for camera control.

3.2 Wide-field to Foveal Transfer

Once an object of interest has been found through the attentional process (explained in Section 4), the cameras are directed so that the object is placed in fixation in the foveal views. This is done using affine transfer (Fairley *et al.*, 1998), which is based on the fact that if the relations between three different views are known, the position of a point given in two views can determined in the third. In our case a new fixation point is found in the wide-field views and the problem is to transfer the same point to each foveal view. To relate a foveal view position $\mathbf{x_f}$ to the corresponding wide-field position $\mathbf{x_w}$, we use the affine epipolar constraint $\mathbf{x_w^\top F_A x_f} = 0$ and the affine essential matrix

$$\mathbf{F_A} = \begin{pmatrix} 0 & 0 & a_3 \\ 0 & 0 & a_4 \\ a_1 & a_2 & a_5 \end{pmatrix} \tag{2}$$

Here a_1, a_2, a_3 and a_4 encode the relative orientation and scale between the wide-field and foveal views, while a_5 is the difference in y-wise position between the optical centers. Similarly to wide-field calibration, the parameters are determined using feature matching of Harris' corner features (Harris and Stephens, 1988) and random sampling (Fischler and Bolles, 1981). With wide-field views related to each other using Equation (1) and foveal views to their wide-field counterparts using Equation (2), a new fixation point can be transferred from the wide-field to the foveal views. The cameras can then be moved using rapid gaze shifts, so that the new point is placed in the center of the foveal images.

3.3 Fixation

Once a transfer has been completed and a saccade (rapid gaze shift) executed towards the new attention point, the system tries to fixate onto the new region in the center of the foveal views. This fixation is kept until another region of interest has been found through the attentional system. Thus the camera control can be in either of two modes, saccade or fixation. However, since a saccade occurs in a fraction of a second, the cameras are almost always in fixation. This is beneficial to more high-level processes. With regions of interest in fixation, binocular information can be extracted, information that can be useful for segmentation, object recognition and manipulation. We will see examples of this in later sections.

The relative orientations of the left and right foveal views are constantly kept up-to-date, much like the wide-field external calibration in Section 3.1. Harris' corner features (Harris and Stephens, 1988) are extracted from both views and features are matched using random sampling (Fischler and Bolles, 1981). The cameras are then related by an affine essential matrix $\mathbf{F_A}$, similar to the one

used for wide-field to foveal transfer in Equation (2). Even if $\mathbf{F_A}$ is just an approximation of a general essential matrix, it is applicable to our case, since focal lengths are large and views narrow. Binocular disparities are measured along the epipolar lines and the vergence angle of the stereo head is controlled such that the highest density of points are placed at zero disparity. For temporal filtering we use Kalman filters, but ignore frames for which not enough matches are found.

4 Bottom-Up and Top-Down Attention

The best way of viewing attention in the context of a robotic system is as a selection mechanisms serving the higher level tasks such as object recognition and manipulation. Biological systems may provide a good basis for solving some of the modeling issues. However, due to computational issues mentioned earlier, these studies serve as a mere inspirational source and should not be restricting the computational implementation. We know that humans tend to perform a subconscious ranking of the "interestingness" of the different components of a visual scene. This ranking depends on the observers goals as well as the components of the scene; how the components in the scene relate to their surroundings (Bottom-up) and to our objectives (Top-down) (Itti, 2000; Li, 2002). In humans, the attended region is then selected through dynamic modifications of cortical connectivity or through the establishment of specific temporal patterns of activity, under both Top-down (task dependent) and Bottom-up (scene dependent) control (Olshausen *et al.*, 1993). In this work we will define the Top-down information as consisting of two components: 1) task dependent information which is usually volitional, and 2) contextual scene dependent information.

We propose a simple, yet effective, Artificial Neural Network approach that learns the optimal bias of the Top-down saliency map (Koch and Ullman, 1985). The most novel part of the approach is a dynamic combination of the Bottom-up and Top-down saliency maps. Here an information measure (based on entropy measures) indicates the importance of each map and thus how the linear combination should be altered over time. The combination will vary over time and will be governed by a differential equation. Together with a mechanism for Inhibition-of-Return, this dynamic system manages to adjust itself to a balanced behavior, where neither Top-down nor Bottom-up information is ever neglected.

4.1 Biased Saliency for Visual Search Tasks

Current models of how the attentional mechanism is incorporated in the human visual system generally assume a Bottom-up, fast and primitive mechanism that biases the observer toward selecting stimuli based on their saliency (most likely encoded in terms of center-surround mechanisms) and a second slower, Top-down mechanism with variable selection criteria, which directs the 'spotlight of attention' under cognitive, volitional control (Treisman and Gelade, 1980). In computer vision, attentive processing for scene analysis initially largely

dealt with salience based models, following (Treisman and Gelade, 1980) and the influential model of Koch and Ullman (1985). However, several computational approaches to selective attentive processing that combine Top-down and Bottom-up influences have been presented in recent years.

Koike and Saiki (2002) propose that a stochastic Winner-Take-All (WTA) enables the saliency based search model to cause the variation of the relative saliency to change search efficiency, due to stochastic shifts of attention. Ramström and Christensen (2004) calculate feature and background statistics to be used in a game theoretic WTA framework for detection of objects. Choi *et al.* (2004) suggest learning the desired modulations of the saliency map (based on the model by Itti *et al.* (1998)) for Top-down tuning of attention. Navalpakkam and Itti (2003) enhance the Bottom-up salience model to yield a simple, yet powerful architecture to learn target objects from training images containing targets in diverse, complex backgrounds. Earlier versions of their model did not learn object hierarchies and could not generalize, although the current model could do that by combining object classes into a more general super class.

Lee *et al.* (2003) showed that an Interactive Spiking Neural Network can be used to bias the Bottom-up processing towards a task (in their case in face detection). However, their model was limited to the influence of user provided Top-down cues and could not learn the influence of context. In Frintrop's VOCUS-model (Frintrop, 2006) there are two versions of the saliency map; a Top-down map and a Bottom-up one. The Bottom-up map is similar to that of Itti and Koch's, while the Top-down map is a tuned version of the Bottom-up one. The total saliency map is a linear combination of the two maps using a fixed user provided weight. This makes the combination rigid and non flexible, which may result in loss of important Bottom-up information. Oliva *et al.* (2003) show that Top-down information from visual context can modulate the saliency of image regions during the task of object detection. Their model learns the relationship between context features and the location of the target during past experience in order to select interesting regions of the image. Many of the computational models study the attention mechanism itself but there have also been approaches that demonstrate robotic applications.

In the work of Breazeal and Scassellati (1999) a computational implementation of the visual search model described by Wolfe (1994), is created. They use this system on a robotic platform where they integrate perception with inhibition of return and other internal effects. The result is a context-dependent attention map they use to determine the gaze direction of the robot. On their humanoid platform, Vijayakumar *et al.* (2001) explored the integration of oculomotor control (biologically inspired) with visual attention. Their attention module consisted of a neural network implementing a WTA-network (Tsotsos *et al.*, 1995), in which different visual stimuli could compete for shift of gaze in their direction. Nickerson *et al.* (1998) created within the framework of the ARK project, mobile robots that used attention-based space segmentation for navigating within industrial environments without the need for artificial landmarks. They too used the WTA-network of (Tsotsos *et al.*, 1995). Clark and Ferrier (1992) suggested early on an robotic implementation of a visual control

system based on models of the human oculomotor control. They did that by attentive control through specifications of gains in parallel feedback loops. Their modal control was through saliency map calculated as weighted combination of several feature maps. In a recent work, Siagian and Itti (2007) use the attention system of Itti *et al.* (1998) in combination with a "gist" model of the scene, to direct an outdoor robot toward the most likely candidate locations in the image, thus making the time-consuming process of landmark identification more efficient. In their human-machine interaction system, Heidemann *et al.* (2004) recognize hand gestures in parallel with computing context-free attention maps for the robot. Allowing an interaction between the human and the robot where, according to the authors, a balanced integration of bottom-up generated feature maps and top-down recognition is made. One of the few recent works that incorporates a computational mechanism for attention into a humanoid platform is the work of Moren *et al.* (2008). A method called Feature Gating is used to achieve Top-down modulation of saliencies.

Our framework is based on the notion of saliency maps (SMs), (Koch and Ullman, 1985). To define a Top-down SM, $SM_{TD}(t)$, t denoting time, we need a preferably simple search system based on a learner that is trained to find objects of interest in cluttered scenes. In parallel, we apply an unbiased version of the same system to provide a Bottom-up SM, $SM_{BU}(t)$. In the following we will develop a way of computing these two types of maps and show that it is possible to define a dynamic active combination where neither one always wins, i.e. the system never reaches a static equilibrium, although it sometimes reaches dynamic one. The model (illustrated in Fig. 4) contains a standard Saliency Map (SM_{BU}) and a Saliency Map biased with weights (SM_{TD}). The Top-down bias is achieved by weight association (our Neural Network). An Inhibition-of-Return mechanism and stochastic WTA-network gives the system its dynamic behavior described in Section 4.3. Finally the system combines $SM_{BU}\,(t)$ and $SM_{TD}\,(t)$ with a linear combination that evolves over time t. Our model applies to visual search and recognition in general, and to cases in which new visual information is acquired in particular.

Several computational models of visual attention have been described in the literature. One of the best known systems is the *Neuromorphic Vision Toolkit* (NVT), a derivative of the model by Koch and Ullman (1985) that was (and is) developed by the group around Itti et al. (Itti and Koch, 2001; Itti *et al.*, 1998; Navalpakkam and Itti, 2003). We will use a slightly modified version of this system for our computations of saliency maps. Some limitations of the NVT have been demonstrated, such as the non robustness under translations, rotations and reflections, shown by Draper and Lionelle (2003). However, our ultimate aim is to develop a system running on a real time active vision system and we therefore seek to achieve a fast computational model, trading off time against precision. NVT is suitable in that respect. Similarly to Itti's original model, we use color, orientation and intensity features, with the modification that we have complemented these with a texture cue that reacts to the underlying texture of regions, not to outer contours. The details of how this texture cue, based on the eigen-values of small patches in the image, are calculated can be found in

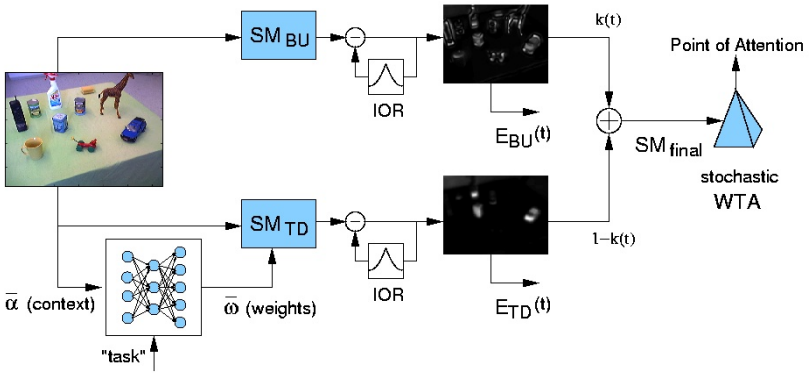


Figure 4: An attentional model that combines Bottom-up and Top-down saliency, with Inhibition-of-Return and a stochastic Winner-Take-All mechanism, with context and task dependent Top-down weights.

(Rasolzadeh *et al.*, 2007).

4.2 Weight Optimization and Contextual Learning

As mentioned above we base both Top-down and Bottom-up salience on the same type of map. However, to obtain the Top-down version we bias this conspicuity map. In our approach, which otherwise largely follows Frintrop (2006), the weighting is done differently. This has important consequences, as it will be shown later. The four broadly tuned color channels R, G, B and Y, all calculated according to the NVT-model, are weighted with the individual weights $(\omega_R, \omega_G, \omega_B, \omega_Y)$. The orientation maps $(O_{0^\circ}, O_{45^\circ}, O_{90^\circ}, O_{135^\circ})$ are computed by Gabor filters and weighted with similar weights $(\omega_{0^\circ}, \omega_{45^\circ}, \omega_{90^\circ}, \omega_{135^\circ})$ in our model. Following the original version, we then create scale pyramids for all 9 maps (including the intensity map I) and form conventional center-surround-differences by across-scale-subtraction and apply Itti's normalization operator. [2] This leads to the final conspicuity maps for intensity $(\bar{I})$, color $(\bar{C})$, orientation $(\bar{O})$ and texture $(\bar{T})$. As a final set of weight parameters we introduce one weight for each of these maps, $(\omega_I, \omega_C, \omega_O, \omega_T)$. To summarize the calculations:

$$RG(c,s) = |(\omega_R \cdot R(c) - \omega_G \cdot G(c)) \ominus (\omega_R \cdot R(s) - \omega_G \cdot G(s))|$$
$$BY(c,s) = |(\omega_B \cdot B(c) - \omega_Y \cdot Y(c)) \ominus (\omega_B \cdot B(s) - \omega_Y \cdot Y(s))|$$
$$O_\theta(c,s) = \omega_\theta \cdot |O_\theta(c) \ominus O_\theta(s)|$$
$$\bar{C} = \bigoplus_c \bigoplus_s N(RG(c,s)) - N(BY(c,s))$$
$$\bar{O} = \sum_\theta N(\bigoplus_c \bigoplus_s N(O_\theta(c,s)))$$

[2] The center-surround-differences are a computational model of the center-surround receptive fields composed by ganglion cells in the retina. For details on the across-scale subtraction we refer to Itti's original work.

$$\bar{I} = \bigoplus_c \bigoplus_s N(|I(c) \ominus I(s)|)$$
$$\bar{T} = \bigoplus_c \bigoplus_s N(|T(c) \ominus T(s)|)$$
$$SM_{TD} = \omega_I \bar{I} + \omega_C \bar{C} + \omega_O \bar{O} + \omega_T \bar{T}$$

Here $\ominus$ denotes the across-scale-subtraction, $\bigoplus$ the across-scale-summation. The center scales are $c \in \{2, 3, 4\}$ and the surround scales $s = c + \delta$, where $\delta \in \{3, 4\}$ as proposed by Itti and Koch. We call the final modulated saliency map the Top-down map, SM_{TD}. The Bottom-up map, SM_{BU} can be regarded as the same map with all weights being 1.

As pointed out by Frintrop, the number of introduced weights in some sense represents the degrees of freedom when choosing the "task" or the object/region to train on. A relevant question to pose is: how much "control" do we have over the Top-down map by changing the weights? As previously stated, we divide Top-down information in two categories; i) task and ii) context information. To tune and optimize the weight parameters of the SM for a certain task, we also have to examine what kind of context information is important. For instance, the optimal weight parameters for the same task typically differ from one context to the other. These two issues will be considered in the remaining part of the section.

4.2.1 Optimizing for the ROI

First we need to formalize the optimization problem. For a given Region Of Interest (ROI) characteristic for a particular object, we define a measure of how the Top-down map differs from the optimum as:

$$e_{ROI}(\bar{\omega}) = \frac{\max SM_{TD} - \max\left(SM_{TD}|_{ROI}\right)}{\max SM_{TD}}$$

where $\bar{\omega} = (\omega_I, \omega_O, \omega_C, \omega_T, \omega_R, \omega_G, \omega_B, \omega_Y, \omega_{0^\circ}, \omega_{45^\circ}, \omega_{90^\circ}, \omega_{135^\circ})$ is the weight vector. The optimization problem will then be given by $\bar{\omega}_{opt} = \arg\min e_{ROI}(\bar{\omega})$. $\bar{\omega}_{opt}$ maximizes peaks within the ROI and minimizes peaks outside ROI. This optimization problem can be solved with the Levenberg-Marquardt algorithm. The optimal set of weights (the optimal weight-vector) are thus obtained. With this set of weights, we significantly increase the probability of the winning point being within a desired region. To summarize; given the task to find a certain (type) of ROI we are able to find a good set of hypotheses by calculating the Top-down map $SM_{TD}(\bar{\omega}_{opt})$. The method used to do this optimization for a given ROI, is described in (Rasolzadeh *et al.*, 2006).

4.2.2 Learning Context with a Neural Network

The weight optimization above is in principle independent of context. In order to include the correlation between the optimal weights and the context (environmental Top-down information), we have to know both types of Top-down information (context and task) in order to derive the set of optimal weights as a function of context and task.

There are a large number of different definitions of context in the computer vision literature (Rabinovich *et al.*, 2007; Strat and Fischler, 1989, 1995). In our model we will keep the definition simple enough to serve our purpose of visual search. A simple example is that a large weight on the red color channel would be favorable when searching for a red ball on a green lawn. However, the same weighting would not be appropriate when searching for the same ball in a red room! We therefore represent context by the total energy of each feature map, in our case a 11-dimensional contextual vector, here denoted as $\bar{\alpha}$. This will give us a notion of "how much" of a certain feature we have in the environment, and thus how discriminative that feature will be for a visual search task.

Obviously we cannot perform this non-convex (time-consuming) optimization every time we need to find the optimal weights (minimizing $e_{ROI}(\bar{\omega})$), in order to find a ROI, i.e. have maximal saliency within the ROI. Instead, we collect the optimized weights and the corresponding contextual vectors for a large set of examples. Given that data set, we train an artificial neural network (Haykin, 1994) to associate between the two: i.e. given a contextual vector, what will the optimal set of weights be like. This is performed for each type of ROI, thus there will be one trained neural network (NN) for each object. Each of these NNs can automatically correlate the context information with the choice of optimal weight parameters *without* any optimization. Fore more details on how this training is done we refer to our previous works (Rasolzadeh *et al.*, 2006).

4.3 Top-Down / Bottom-Up Integration

So far we have defined a Bottom-up map $SM_{BU}(t)$ representing the unexpected feature based information flow and a Top-down map $SM_{TD}(t)$ representing the task dependent contextual information. We obtain a mechanism for visual attention by combining these into a single saliency map that helps us to determine where to "look" next.

In order to do this, we rank the "importance" of saliency maps, using a measure that indicates how much value there is in attending that single map at any particular moment. We use an energy measure (E-measure) similar to that of Hu et al, who introduced the *Composite Saliency Indicator* (CSI) for similar purposes (Hu *et al.*, 2004). In their case, however, they applied the measure on each individual feature map. We use the same measure, yet we use it on the summed up saliency maps. The Top-down and Bottom-up energies E_{TD} and E_{BU} are defined as the density of saliency points divided by the convex hull of all points.

Accordingly, if a particular map has many salient points located in a small area, that map might have a higher E-value than one with even more salient points, yet spread over a larger area. This measure favors saliency maps that contain a small number of very salient regions.

4.3.1 Combining SM_{BU} and SM_{TD}

We now have all the components needed to combine the two saliency maps. We may use a regulator analogy to explain how. Assume that the attentional system contains several (parallel) processes and that a constant amount of processing power has to be distributed among these. In our case this means that we want to divide the attentional power between $SM_{BU}(t)$ and $SM_{TD}(t)$. Thus the final saliency map will be a linear combination

$$SM_{final} = k \cdot SM_{BU} + (1 - k) \cdot SM_{TD}.$$

Here the k-value varies between 0 and 1, depending on the relative importance of the Top-down and Bottom-up maps, according to the tempo-differential equation

$$\frac{dk}{dt} = -c \cdot k(t) + a \cdot \frac{E_{BU}(t)}{E_{TD}(t)} \quad , \quad 0 \leq k \leq 1$$

The two parameters c and a, both greater than 0, can be viewed as the amount of *concentration* (devotion to search task) and the *alertness* (susceptibility for Bottom-up info) of the system. The above equation is numerically solved between each attentional shift.

The first term represents an integration of the second one. This means that a saliency peak needs to be active for a sufficient number of updates to be selected, making the system less sensitive to spurious peaks. If the two energy measures are constant, k will finally reaches an equilibrium at aE_{BU}/cE_{TD}. In the end, SM_{BU} and SM_{TD} will be weighted by aE_{BU} and $\max(cE_{TD} - aE_{BU}, 0)$ respectively. Thus the Top-down saliency map will come into play, as long as E_{TD} is sufficiently larger than E_{BU}. Since E_{TD} is larger than E_{BU} in almost all situations when the object of interest is visible in the scene, simply weighting SM_{TD} by E_{TD} leads to a system dominated by the Top-down map.

The dynamics of the system comes as a result of integrating the combination of saliencies with Inhibition-of-Return (IOR). The kind of IOR we talk about here is in a covert mode, where the eyes or the head are not moving at all (overt shifts) and there is essentially only a ranking of the various saliency peaks within the same view. Of course, if the desired object is not found within the current set of salient points, the system will continue to an overt shift where the head and the eyes focus on a different point in space. If a single salient Top-down peak is attended to (covertly), saliencies in the corresponding region will be suppressed, resulting in a lowered E_{TD} value and less emphasis on the Top-down flow, making Bottom-up information more likely to come into play. However, the same energy measure will hardly be affected if there are many salient Top-down peaks of similar strength. Thus the system tends to visit each Top-down candidate before attending to purely Bottom-up ones. This, however, depends on the strength of each individual peak. Depending on *alertness*, strong enough Bottom-up peaks could just as well be visited first. The motivation for a balanced combination of the two saliency-maps based on two coefficients

Figure 5: Disparity map (right) of a typical indoor scene (left).

(*alertness* and *concentration*) comes from neuroscientific studies on the dorsal and ventral pathways. In a larger system, this attentive mechanism we propose here can thus easily be integrated with the rest of the reasoning system, with the dorsal stream activity indicating a pragmatic mode, whereas the the ventral stream activity indicates a semantic mode.

4.4 Binocular Cues for Attention

Since the attentional system described above is generic with respect to the visual task, it may just as well deliver regions of interest corresponding to things that are either too large or too far away to be manipulated. It is clear that in our scenario, size and distance needs to be taken into account for successful interaction with the environment. Now, even if the projective size of a region can be measured, its real-world size is unknown, since the projective size depends on the distance from the camera set. One of the benefits of a binocular system, such as the one described in Section 3, is that sizes and distances can be made explicit. Therefore, we complement the attentional system with binocular information in order to make the system more likely to pop-out regions of interest suitable for manipulation.

With wide-field cameras calibrated as described in Section 3.1 disparity maps, such as the one to the right in Fig. 5, are computed. Disparity maps encode distances to 3D points in the scene. A point distance is given by $Z = bf/d$, where b is the baseline (the distance between the cameras), f is the focal length and d the respective binocular disparity. Before a peak is selected from the saliency map, the saliency map is sliced up in depth into a set of overlapping layers, using the disparity map. Each layer corresponds to saliencies within a particular interval in depth. A difference of Gaussian (DoG) filter is then run on each layer. The sizes of these filters are set to that of the expected projected sizes of manipulable things. Thus for saliency layers at the distance the DoGs are smaller than for layers closer to the camera head. As a result you will get saliency peaks similar to those in Fig. 6, with crosses indicating the expected size of things in the scene.

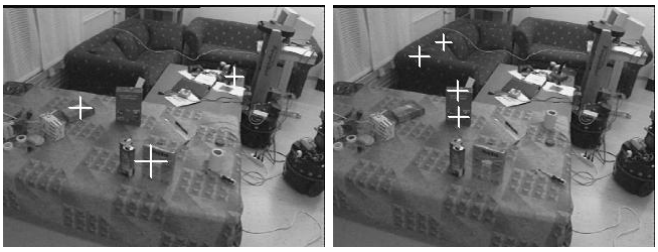


Figure 6: Saliency peaks with saliency maps computed using top-down tuning for the orange package (left) and the blue box (right). The crosses reflect the sizes derived from the attentional process.

Figure 7: Disparity maps (right), prior foreground probabilities (middle) and posteriori figure-ground segmentation (left).

5 Foveated Segmentation

After a region of interest has been selected by the attentional system, the camera system is controlled such that the region is placed at zero disparity in the center of the foveal views. It is now ready to be identified and possibly manipulated. Before this is done, it would be beneficial if it could also be segmented from other nearby objects in the scene. Both recognition and pose estimation are simplified if the object is properly segmented. In our system we do this with the help of binocular disparities extracted from the foveal views.

In the foveated segmentation step, the foreground probability of each pixel is computed in a probabilistic manner. From area based correlation we estimate a measurement for each pixel, that are then used to estimate the prior probability of a pixel belonging to the foreground. Examples of foreground priors can be seen in Fig. 7 (middle).

By modeling the interaction between neighboring patches and computing the posteriori foreground probabilities using graph-cuts, pixels are finally labeled as being part of either the *foreground* or *background*. Since there are only two possible labels, the exact posteriori solution is given in a single graph-cut operation

(Grieg *et al.*, 1989). The resulting segmentation may look like the two images in Fig. 7 (right). These segmentations are then passed to either recognition or pose estimation. For more information on the precise modeling and motivations see (Björkman and Eklundh, 2006).

5.1 From 3D Segments to Shape Attributes

In order to have segmentation that is appropriate for manipulation image data needs to be grouped into regions corresponding to possible objects in the 3D scene. Disparities can be considered as measurements in 3D space, clustered around points of likely objects. These clusters are found by applying a kernel-based density maximization method, known as Mean Shift (Comaniciu and Meer, 2002). Clustering is done in image and disparity space, using a 3D Gaussian kernel with a size corresponding to the typical 3D size of objects that can be manipulated. The maximization scheme is iterative and relies on initial center point estimates. As such estimates we use the hypotheses from the attentional system. Examples of segmentation results using this approach can be seen in the second row of Fig. 9.

One major drawback of the mean shift algorithm is the fact that an object can not be reliably segregated from the surface it is placed on, if there is no evidence supporting such a segregation. Without any additional assumption on surface shape or appearance there is no way of telling the surface from the object. However, in many practical scenarios (including ours) it might be known to the robotic system that objects of interest can in fact be expected to be located on flat surfaces, such as table tops.

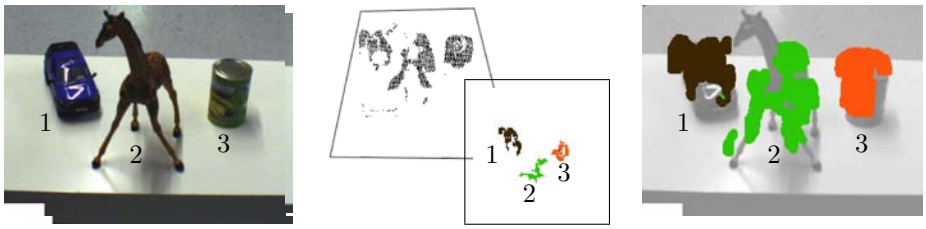
We therefore complement our approach with a table plane assumption. Using a textured surface, it is possible to find the main plane and cut it with a 3D version of the Hough transform as in (Huebner *et al.*, 2008a). Following the table assumption the 3D points are mapped onto a 2D grid to find segments and basic shape attributes.

The result of transformation and clipping on the scene given in Fig. 8(a) can be seen in Fig. 8(b). The segmentation of objects is computed on the 2D grid (Fig. 8(c)) with a simple region growing algorithm grouping pixels into larger regions by expanding them bottom up. Since the grid is thereby segmented, simple shape-based attributes of each segment can be determined and the segments reprojected to 3D points or to the image plane (illustrated in Fig. 8(d)) [3].

5.2 Associated Attributes

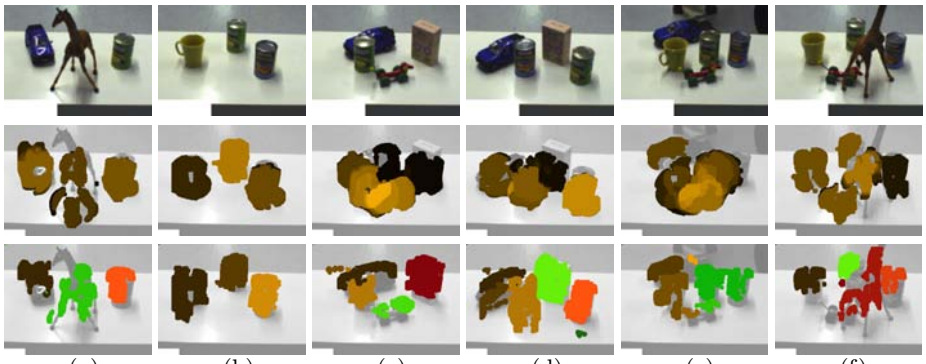
The generated segments are just *things*, as the step to an *object* longs for semantics. One way to identify the semantics of a thing in order to derive an object

[3]Note that dilation has been applied for the reprojected segments for the later application of point-based object hypotheses verification. The dilation, the grid approach, as also noisy and incomplete data from stereo causing that reprojections are often little larger or not completely covering the bodies.



(a) Stereo images (b) 3D points (c) 2D segmentation (d) Reprojection

Figure 8: Segmentation using the table plane assumption. Disparity information from the stereo images (a) produces 3D points (b). Having defined the dominant plane, the points can be projected onto this plane, where distinctive segments are computed (c) and reprojected to the image (d).



(a) (b) (c) (d) (e) (f)

Figure 9: Sample scenario segmentation (best viewed in color). Original images are shown in the first row. The second row shows results using the Mean Shift segmentation, the bottom row those using the table plane assumption (mentioned in Section 5.1). In the latter, (a) and (b) seem well segmented and in (c) there is just some noise at the table edge. Problems arise for (d)-(f): (d) two segments for the car, (e) one segment for two cans, and (f) the dog underneath the giraffe is not detected.

is to associate attributes to it. The attributes can be of two kinds, *intrinsic* and *extrinsic*. Intrinsic attributes are object-centered and thereby theoretically viewpoint-independent (e.g. shape, color, mass). Extrinsic attributes describe the viewpoint-dependent state of an object (e.g. position, orientation). In our system, the basic intrinsic attributes of covered area, length (along the dominant axis), width (perpendicular to the dominant axis) and height can be qualitatively determined for each segment. The discretization, i.e. if an object is *small* or *large* in size, is adapted to our table-top manipulation scenario at hand. Additionally, the centroid position of a segment is calculated. Its 3D point cloud is kept available for the subsequent operations, e.g. pose estimation (as we will show later in Section 6.2) or shape approximation and grasping, as we proposed in (Huebner *et al.*, 2008b).

6 Object Manipulation

To achieve real cognitive capabilities, robotic systems have to be able to interact with the surrounding. Grasping and manipulation of objects is one of the basic building blocks of such a system. Compared to humans or primates, the ability of today's robotic grippers and hands is surprisingly limited and their dexterity cannot be compared to human hand capabilities. Contemporary robotic hands can grasp only a few objects in constricted poses with limited grasping postures and positions.

Grasping, as a core cognitive capability, has also been posed as one of the key factors of the evolution of the human brain. This is founded in convergent findings of brain researchers. For example, 85% of axons in visual cortex do not come from the retina, but other brain areas including what is thought to be higher brain regions (Sigala and Logothetis, 2002). Lately, anatomical and physiological investigations in non human primates, together with brain imaging studies in humans, have identified important cortical pathways involved in controlling visually guided prehension movements. In addition, experimental investigations of prehension movements have begun to identify the sensorimotor transformations and representations that underlie goal directed action. It has been shown that attentional selection of the action related aspects of the sensory information is of considerable importance for action control, (Castiello, 2005; Riddoch *et al.*, 2001). When a grasp is being prepared, the visual system provides information about the egocentric location of the object, its orientation, form, size, and the relevant environment. Attention is particularly important for creating a dynamic representation of peripersonal space relevant for the specific tasks.

Regarding implementation in robots, grasp modeling and planning is difficult due to the large search space resulting from all possible hand configurations, grasp types, and object properties that occur in regular environments. The dominant approach to this problem has been the model based paradigm, in which all the components of the problem (object, surfaces, contacts, forces) are modeled according to physical laws. The research is then focused on grasp analysis, the study of the physical properties of a given grasp; and grasp synthesis, the computation of grasps that meet certain desirable properties, (Bicchi and Kumar, 2000; Coelho Jr. *et al.*, 1998; Namiki *et al.*, 2003; Platt Jr. *et al.*, 2002; Shimoga, 1996). More recently, it has been proposed to use vision as a solution to obtain the lacking information about object shapes or to use contact information to explore the object (Kragic *et al.*, 2005; Morales *et al.*, 2001; Platt Jr. *et al.*, 2002). Another trend has focused on the use of machine learning approaches to determine the relevant features that indicate a successful grasp (Coelho *et al.*, 2001; Kamon *et al.*, 1998; Morales *et al.*, 2004). Finally, there have been efforts to use human demonstrations for learning grasp tasks (Ekvall and Kragic, 2004).

One of the unsolved problems in robot grasping is grasping of unknown objects in unstructured scenarios. For general settings, manipulation of unknown objects has almost not been pursued in the literature and it is commonly as-

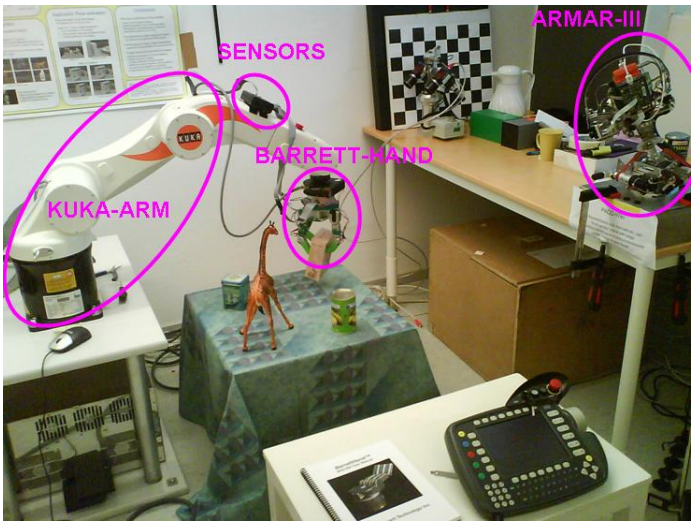


Figure 10: Our robotic setup.

sumed that objects are easy to segment from the background. In the reminder of this section, we concentrate on an example of how the presented visual system can be used to provide grasping hypotheses of objects for which the identity/geometry is not known in advance. We acknowledge that this approach is not valid in all situations, but it is one of the possible directions to pursue.

6.1 Experimental Platform

Our robotic setup consist of the Armar-III humanoid head described in Section 2.1, a BH8-series BarrettHand mounted on a KUKA KR5 R850 6-DOF robot, Fig. 10. The hand-yye calibration is performed using the classical approaches of (Shiu and Ahmad, 1989; Tsai and Lenz, 1988). The integration of the different parts of this robotic platform is achieved using a modularized software system; containing interacting modules for frame grabbing, camera calibration, visual front end modules, head control, arm control, hand control and sensory reading. Modules are implemented as CORBA processes that run concurrently and generally on different machines.

6.2 Model-Free Manipulation

In general, we will not have a precise geometrical model for all objects the robot is supposed manipulate. One a new object hypothesis is made based on the visual modules described so far, different attributes may be attached to it. These attributes are intrinsic (size, shape) and extrinsic (pose) and are stored as a part of object representation for later indexing. We refer to (Huebner *et al.*, 2008a) for more details.

Figure 11: Left) A left manipulation camera image, Middle) The corresponding disparity map, Right) Segmentation from mean shift in 3D space.

Before the 3D position of an object, as well as its orientation can be determined, it has to be segmented from its surrounding, which in our system is done using a dense disparity map as explained in Section 5, and exemplified by the images in Fig. 11. In the current system, we thus use the generated object hypotheses in combination with the orientation estimation described below, to apply top-grasps on the objects. Given the segmentation (with table-plane assumption), and 3D coordinates, a plane is mapped to the 3D coordinates of all points within the segmented object. Since only points oriented toward the cameras are seen, the calculated orientation tends to be somewhat biased toward fronto-parallel solutions. However, the BarrettHand is able to tolerate some deviations from a perfectly estimated orientation. With the 3D points denoted by $\mathbf{X}_i = (X_i, Y_i, Z_i)^\top$, we iteratively determine the orientation of a dominating plane using a robust M-estimator. The normal of the plane at iteration k is given by the least eigenvector $\mathbf{c}_k$ of

$$\mathbf{C}_k = \sum_i \omega_{i,k}(\mathbf{X}_i - \bar{\mathbf{X}}_\mathbf{k})(\mathbf{X}_i - \bar{\mathbf{X}}_k)^\top, \tag{3}$$

where the weighted mean position is $\bar{\mathbf{X}}_\mathbf{k}$.

Points away from the surface are suppressed through the weights

$$\omega_{i,k} = t^2/(t^2 + \delta_{i,k}^2), \tag{4}$$

where $\delta_{i,k} = \mathbf{c}_{k-1}^\top(\mathbf{X}_i - \bar{\mathbf{X}})$ is the distance from the point $\mathbf{X}_i$ to the plane of the previous iteration. Here t is a constant reflecting the acceptable variation in flatness of the surface and is set to about a centimeter. More details on the implementation can be found in (Kragic *et al.*, 2005).

7 Experimental Results

The following sections present several experiments related to the different aspects of the vision system. Section 7.1 presents qualitative and quantitative experiments of the attention system, such as the weight optimization process and the neural network learning. Section 7.2 presents results on the object recognition module and Section 7.3 gives an example of the integrated modules solving an object detection and manipulation task.

7.1 Top-Down and Bottom-Up Attention

As described in Section 4, our attentional model consists of three main modules:

- The optimization of Top-down weights (offline);
- The Neural Network which associates context and weight (online); and
- The dynamical combination of SM_{BU} and SM_{TD}.

The experiments presented below are designed to show how these different modules affect the performance of the model. We present results from the experiments on the contextual learning, since it is the most crucial part for our visual search tasks. In Fig. 12 (top), the ten objects used in the experiments are shown. These are all associated with a set of intrinsic attributes that consist of 3D size, appearance, and feasible grasps. To represent the appearance, we use SIFT descriptors (Lowe, 1999) and color histograms (Gevers and Smeulders, 1999). Detection experiments using these can be found in Section 7.2. The graph in Fig. 12 shows the Top-down (TD) weights deduced for the four cues from one particular image. The cues with high weights for most materials are color and texture. We can see that some cues are almost completely suppressed for several objects. The resulting set of triplets $\{ROI, \bar{\omega}_{opt}, \bar{\alpha}\}$ were used for training the neural networks.

7.1.1 Weight Optimization

The non-convex optimization, solved with the Levenberg-Marquardt method, maximizes the saliency value SM_{TD} within the RIO. RIO represents the desired target object and the process of optimization is based on manipulating the weight-vector $\bar{\omega}$. However, it is important to note that, even if one may reach a global minimum in the weight optimization, it does not necessarily mean that our Top-down map is "perfect", as in Fig. 13. In other words, the Top-down map may not rank the sought ROI the highest, in spite of $e_{ROI}(\bar{\omega})$ being at its global minimum for that specific image and object. What this implies is that for some objects $\min[e_{ROI}(\bar{\omega}_{opt})] \neq 0$, or simply that our optimization method failed to find a proper set of weights for the Top-down map at the desired location as, for example, in Fig. 14.

Another observation worth mentioning is the fact that there may be several global optima in weight space each resulting in different Top-down maps. For example, even if there exists many linear independent weight vectors $\bar{\omega}_i$ for which $e_{ROI}(\bar{\omega}_i) = 0$, the Top-down maps $SM_{TD}(\bar{\omega}_i)$ will in general be different from one another (with different E_{CSI}-measure).

7.1.2 Artificial Neural Network (ANN) Training

When performing the pattern association on the neural network that is equivalent with context learning, it is important that the training data is "pure". This means that only training data that gives the best desired result should be

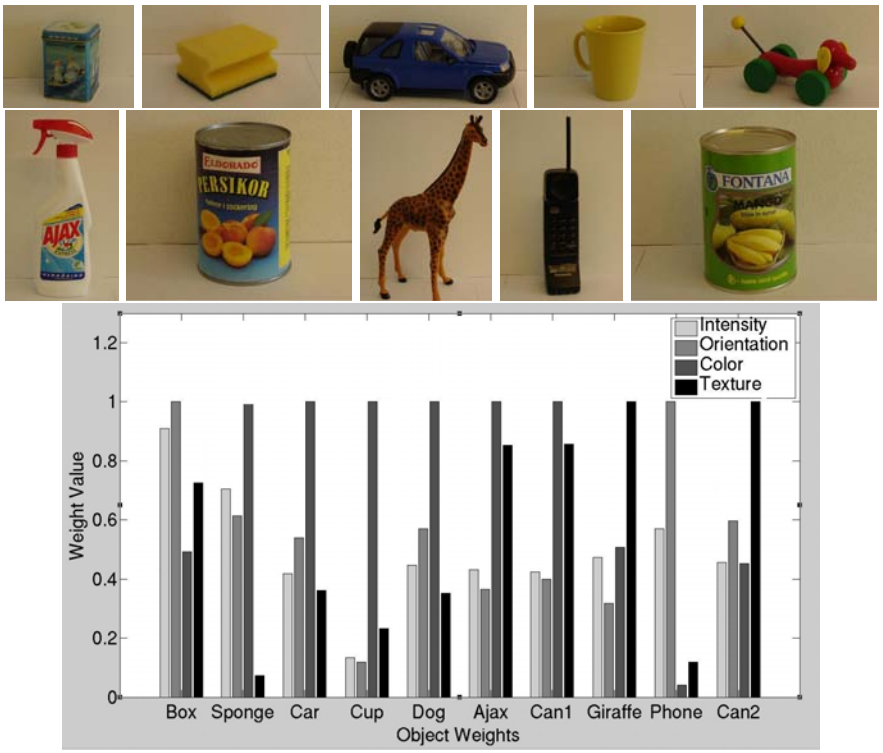


Figure 12: A set of objects used for experiments (left) and the four TD-weights $\bar{\omega}_I, \bar{\omega}_O, \bar{\omega}_C, \bar{\omega}_T$ for each object in one particular image (right).

included. Thus only examples $\{ROI, \bar{\omega}_{opt}, \bar{\alpha}\}$ where $e_{ROI}(\bar{\omega}_{opt}) = 0$ were used. To examine the importance of our context information we created another set of NNs trained without any input, i.e. simple pattern learning. For the NN calculations this simply leads to an averaging network over the training set $\{ROI, \bar{\omega}_{opt}\}$. Quantitative results of these experiments are shown in Fig. 15. There were from a training set of 96 images taken of 10 different objects on 4 different backgrounds (table-cloths) in two different illumination situations. In each of the 96 images, the location of each of the 10 objects is annotated, thus yielding 960 annotated locations (ROIs). See database online (Rasolzadeh, 2006). Results using optimized weights (last row) in some sense represent the best performance possible, whereas searches using only the Bottom-up map perform the worst. One can also observe the effect of averaging (learning weights without context) over a large set; you risk to *always* perform poor, whereas if the set is smaller you may at least manage to perform well on the test samples that resemble some few training samples. Each NN had the same structure, based on 13 hidden neurons, and was trained using the same number of iterations. Since all weights (11) can be affected by all context components (9) and

Figure 13: An example of successful optimization; the ROI is marked in the left image. Without optimization (unitary weights) the saliency map is purely Bottom-up (middle). However, an optimization that minimizes $e_{ROI}(\bar{\omega})$ (in this case to 0) the optimal weight vector $\bar{\omega}_{opt}$ clearly ranks the ROI as the best hypothesis of the Top-down map (right).

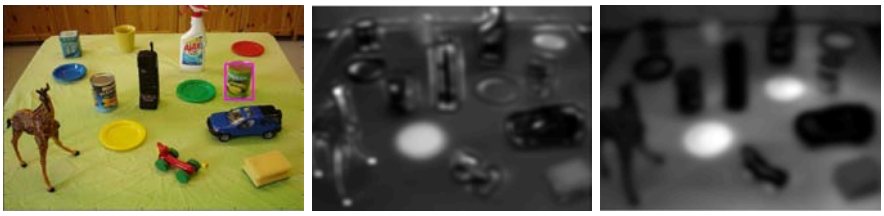


Figure 14: An example of poor optimization; although the optimization may reach a global minimum for $e_{ROI}(\bar{\omega})$ (in this case >0) the optimal weight vector $\bar{\omega}_{opt}$ *doesn't* rank the ROI as the best hypothesis of the Top-down map (right).

since each weight can be increased, decreased or neither, a minimum number of 12 hidden units is necessary for good learning.

7.2 Multi-Cue Object Detection and Hypotheses Validation

Relying on a single object detection method is difficult due to a large variety of objects a robot is expected to manipulate is difficult. Using a combinations of methods seems therefore as a suitable strategy. Without providing an extensive study of all possible methods and combinations, we give an example that shows the benefit of foveated segmentation and multiple cues object recognition. For this purpose, we have selected two methods that show different strengths and weaknesses. The first method is based on color histograms (Gevers and Smeulders, 1999) and the other on scale and rotation invariant SIFT features (Lowe, 1999). Histogram based methods are suited for both uniformly colored and textured objects, but tend to be problematic when objects are not easy to distinguish from the background. Feature based method, work well in cluttered environments, but break down when too few features are extracted due to limited texture.

We selected a set of 24 objects, similar to those in Fig. 12. We performed

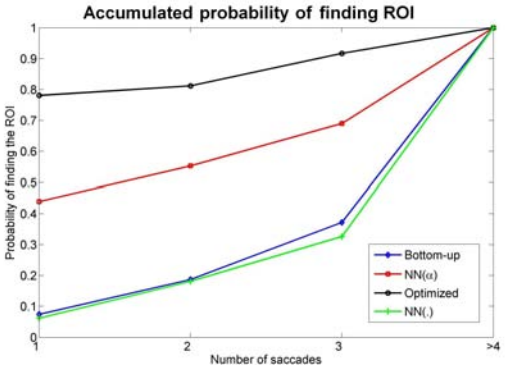


Figure 15: The estimated accumulated probability of finding the ROI. The results were averaged over the entire test set of objects(ROI:s). BU is purely Bottom-up search, $NN_i(\bar{\alpha})$ is Top-down search guided by a Neural Network (trained on i% of the training data available) choosing context dependent weights, and $NN_i(.)$ is the same without any context information.

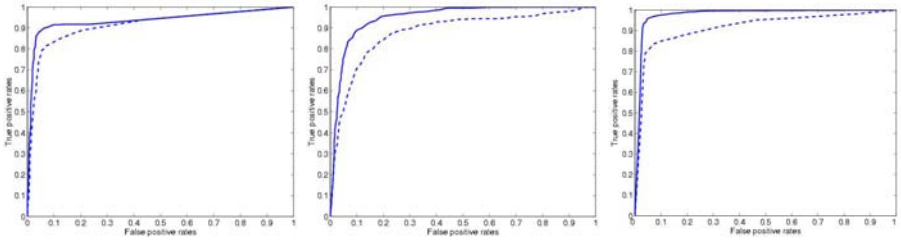


Figure 16: ROC curves for SIFT based (left), color histogram based (middle) and combined (right) object detection, with (solid) and without (dashed) foveated segmentation.

886 object recognition tasks using images provided in real-time using the binocular attention system described in earlier sections. The ROC curves in Fig. 16 illustrate the recognition performance with and without segmentation for both methods individually, as well as for a combination. The combination is done using a binary operator that is learned using a support vector machine (SVM) approach, (Björkman and Eklundh, 2006).

Since we are also interested in object manipulation, we combine the results of appearance and shape recognition where the shape here is represented by the width, breadth and height of the object. Thus, we bind the object identity to its known intrinsic attributes. This binding serves two purposes: i) it boosts the recognition rate by disregarding more false positives, ii) it allows for substitution of objects with other "visually similar" objects. This opens up for broader Object-Action-Complex (OAC) categorization of objects and is discussed further in (Huebner *et al.*, 2008a) as in more detail in (Geib *et al.*,

2006; Kraft *et al.*, 2008; Wörgötter *et al.*, 2009). Since "action" here implies possible (stable) grasps, this binding of identity with intrinsic attributes leads to a scenario where objects that resemble each other (in appearance and shape) may be grasped similarly.

7.3 Object Grasping: an Example

Several object grasping trials were performed and the overall performance of the system was qualitatively evaluated in a tabletop scenario. The goal for the robot was to find a desired object or object type and move it to a predefined location. The first task is to find the object of interest. Here the attention system was tuned by our NN, that selected appropriate weights for the SM_{TD} based on task (i.e. object) and context (scene). That gave us hypotheses of where the object of interest might be. Fig. 17 shows two such examples of SM_{TD} when searching for the 'UncleBen' object and the 'yellowCup' object, respectively. Given any of these hypotheses of location, a saccade was performed to redirect the robot's focus to that particular point in the environment. Consequently the binocular system tried to fixate on that point by the fixation mechanisms described earlier.

Next, a segmentation based on disparities, using the table-plane assumption mentioned in Section 5.1, was made on the "thing" of interest. Segmentation results can be viewed as the enclosed regions in the foveal views of the four examples in Fig. 18. One consequence of real world conditions such as noise, varying illumination etc., is that the segmentation are far from perfect. However, following the OAC-concept mentioned earlier, it is not our goal to gather information about the state of the object solely through vision. Instead we want to complement this sensory information through interactions with the object. Therefore, this imperfection is of minor importance, if the grasping yields a successful result.

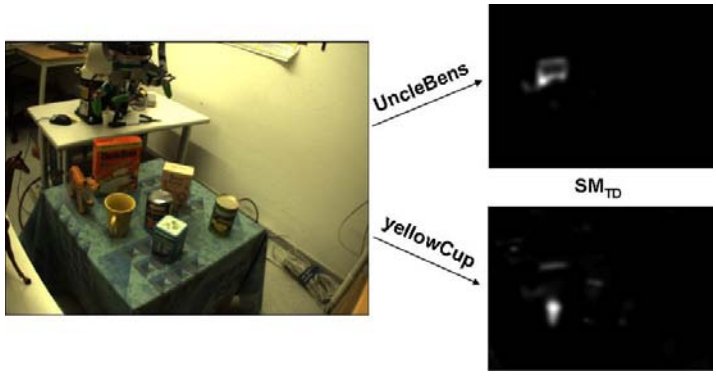


Figure 17: Example with Top-down tuned saliency maps (UncleBens & yellowCup)

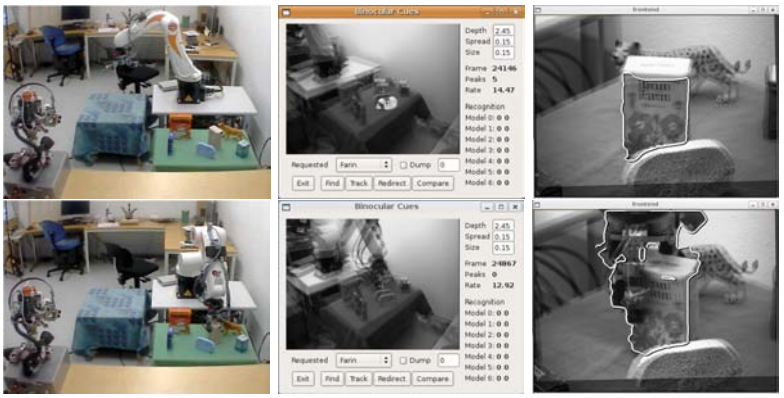
Figure 18: The visual front-end. The top row shows the wide-field view where the visual search selection is made. The bottom row shows the foveal view in which the binocular segmentation and recognition as well as validation is done.

If the segmented region contains the object sought for based on the appearance and intrinsic attributes, the estimated position and orientation is sent to the manipulator. The system then chooses an appropriate grasp based on the intrinsic and extrinsic attributes of the object.

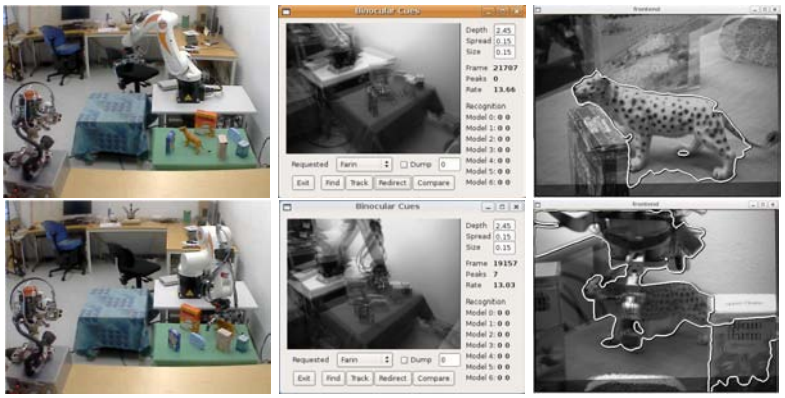
A couple of examples are shown in Fig. 19. The images show the scene before (top rows) and during grasping (bottom rows). One interesting detail seen in these images, is that when the gripper enters the foveal view the fixation-loop adapts to its presence and tries to re-fixate on the point in the center of the image, now being closer to the eyes.

One important detail about this particular implementation is that we have here not included the Bottom-up cues (SM_{BU}) nor the temporal linear combination of the two saliency maps. The reason for this is simply that we were only interested in the Top-down performance of the system. The more dynamic combination of the two saliency maps will be further examined in our future work, where a more "natural" environment with clutter and distractors that might be of importance, will be used.

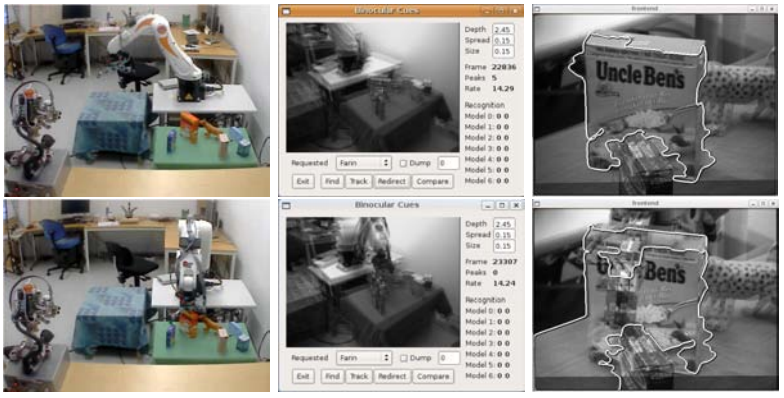
Imperative in the context is that this is just one example to expose the qualitative properties of the system. A potential quantitative and objective evaluation can be difficult to perform for the complex real-world applications that we are facing. Thus other than a separated part-wise evaluation of the different components of this assembled system, we will not here present any quantitative performance results. However, we do intend to create such evaluation processes in the future to more exactly measure the performance of the system as a whole.



(a) Farin



(b) Tiger



(c) UncleBen

Figure 19: Finding and manipulating three different objects. In each of the three examples, the top row shows the state of the system before grasping and the bottom row shows the attempted grasp. Best viewed in color.

8 Conclusions

The goal for the future development of intelligent, autonomous systems is to equip them with the ability to achieve cognitive proficiency by acquiring new knowledge through interaction with the environment and other agents, both human and artificial. The base for acquiring new knowledge is the existence of a strong perception-action components where flexible and robust sensing plays a major role. Visual sensing has during the past few years proved to be the sensory modality that offers the richest information about the environment. Despite this it has typically been used for well defined, specific tasks for the purpose of coping with the complexity and noise effects.

For the past several years, our work has concentrated on the development of general systems and their applications in navigation and object manipulation applications. The work presented here is in line with the development of such a system, except that we have kept our attention on the design and development of a vision system architecture that allows for more general solutions in service robot settings.

Our system uses binocular cues extracted from a system that is based on two sets of cameras: a wide field for attention and a foveal one for recognition and manipulation. The calibration of the system is performed online and facilitates the information transfer between the two sets of cameras. The importance and role of Bottom-up and Top-down attention is also discussed and shown how biased saliency for visual search tasks can be defined. Here, intensity, color, orientation and texture cues facilitate the context learning problem. The attentional system is then complemented with binocular information to make the system more likely to pop out regions of interest suitable for manipulation. We have also discussed how the attentional system can adapt to context changes.

In relation to manipulation, we show and discuss how the system can be used for manipulation of objects for which geometrical model is not known in advance. Here, the primary interest is to pick up an object and retrieve more information about it by obtaining several new views. Finally, we present experimental results of each, and give an example of how the system has been used in one of the object pick-up scenarios. As mentioned, this was just one example to expose the qualitative properties of the system. In real-world applications it is in general difficult to perform extensive experiments thus evaluation different modules in a number of benchmarking tasks may be one of the solutions. Current directions in the area of robotics and different competitions of mobile and manipulation settings are pointing in this direction. However, there are still very few systems that use active vision.

Regarding the limitations of the presented systems we first need to touch upon the issue of using four-camera setup. As we discussed, the ability to use wide-field and narrow-field cameras is good but it is not necessary in all applications. The area of robot navigation and localization, which is currently going more into direction of using visual sensing, may not necessary require such a setup. In addition, one may argue that if a camera system is placed on a mobile base, the robot can move toward the object for achieving a better

view of the object. Our opinion is still that changing between the cameras may be faster and alleviates the need for obstacle avoidance and path planning that moving a platform commonly requires.

Another aspect is the comparison with a zooming camera system. Our opinion here is that changing zoom and fixating on a target results in loosing the wide-field coverage that may be necessary when re-detection of objects for tracking is attempted.

An interesting research issue is related to further learning of object attributes and affordances. The affordances need to be meaningful and related to tasks a robot is expected to execute. In addition, using a interactive setup where a robot can grasp objects, offers more freedom in terms of what attributes can be verified, e.g. hollowness, or extracted, e.g. weight.

The current system does not perform any long-term scene representation, i.e. there is no real memory in the system apart from storing the individual object's attributes. One aspect of future research is therefore to investigate large-scale spatial/temporal representations of the environment. Some aspects of our previous work in the area of Simultaneous Localization and Mapping as well as semantic reasoning will be exploited here.

In the current work, we study the issue of calibration between the head cameras but the hand-eye calibration is not really tackled. The system is complex so that there are also neck motions that could be taken into account for online learning of the hand-head-eye calibration. This process is also related to the issue of smooth pursuit once moving objects are considered. An application may be just the classical object tracking or observation of human activities. An active vision system allows for fixation on parts of human that are important for the task at hand: fixating on mouth when a human is peaking or fixating on the hands when a human is manipulating objects. In this case, the interplay between the saccades and the neck motions is an interesting problem and the solution can be biologically motivated: fast movement of eyes followed by a slower movement of the neck and the mutual compensation.

One of the aspects not studied is vision based closed-loop control of arm motions: visual servoing. It would be interesting to explore, similarly to the partitioned control approaches based on the integration of image based and position based control, to what extent the change between using one of the four cameras at the time can cope with the problems of singularities and loss of features that are inherit to the image based and position based visual servoing approaches.

The most immediate extension of the system is the integration of the object attributes that are extracted based purely on visual input and the ones that are further extracted once the object has been picked up by the robot arm. These include both more detailed visual representation such as several views of the unknown object and the attributes that are extracted by other sensors on the robot: force-torque sensor in the wrist and haptic sensors on the fingers of the robot hand.

Appendix A: Index to Multimedia Extensions

The multimedia extensions to this article can be found online by following the hyperlinks from www.ijrr.org.

Table 1: Index to Multimedia Extensions

Extension	Media Type	Description
1	Video	The grasping experiment described in Section 7.3.
2	Images	Large size multipage(6)-Tiff of Fig. 19(a).
3	Images	Large size multipage(6)-Tiff of Fig. 19(b).
4	Images	Large size multipage(6)-Tiff of Fig. 19(c).

Acknowledgments

This work has been supported by EU through the project PACO-PLUS, FP6-2004-IST-4-27657 and GRASP, IST-FP7-IP-215821.

References

Asfour, T., Regenstein, K., Azad, P., Schroder, J., Bierbaum, A., Vahrenkamp, N., and Dillmann, R. (2006). ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 169–175.

Bicchi, A. and Kumar, V. (2000). Robotic Grasping and Contact: A Review. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 348–353.

Björkman, M. and Eklundh, J.-O. (2002). Real-Time Epipolar Geometry Estimation of Binocular Stereo Heads. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **24**(3), 425–432.

Björkman, M. and Eklundh, J.-O. (2006). Vision in the real world: Finding, attending and recognizing objects. *International Journal of Imaging Systems and Technology*, **16**(5), 189–209.

Björkman, M. and Kragic, D. (2004). Combination of Foveal and Peripheral Vision for Object Recognition and Pose Estimation. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 5, pages 5135–5140.

Breazeal, C. and Scassellati, B. (1999). A Context-Dependent Attention System for a Social Robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1146–1153. Morgan Kaufmann Publishers Inc.

Castiello, U. (2005). The Neuroscience of Grasping. *Nature Neuroscience*, **6**, 726–736.

Choi, S., Ban, S., and Lee, M. (2004). Biologically Motivated Visual Attention System using Bottom-Up Saliency Map and Top-Down Inhibition. *Neural Information Processing - Letters and Review*, **2**, 19–25.

Clark, J. J. and Ferrier, N. J. (1992). Attentive Visual Servoing. In *Active Vision*, pages 137–154. MIT Press.

Coelho, J., Piater, J., and Grupen, R. (2001). Developing Haptic and Visual Perceptual Categories for Reaching and Grasping with a Humanoid Robot. *Robotics and Autonomus Systems*, **37**, 195–218.

Coelho Jr., J., Souccar, K., and Grupen, R. (1998). A Control Basis for Haptically-Guided Grasping and Manipulation. Technical Report CMP-SCI Technical Report 98-46, Dept. Computer Science, University of Massachusetts.

Comaniciu, D. and Meer, P. (2002). Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5), 603–619.

Draper, B. and Lionelle, A. (2003). Evaluation of Selective Attention under Similarity Transforms. In *Proc. International Workshop on Attention and Performance in Computer Vision*, pages 31–38.

Ekvall, S. and Kragic, D. (2004). Interactive Grasp Learning Based on Human Demonstration. In *IEEE/RSJ International Conference on Robotics and Automation*, pages 3519–3524.

Ekvall, S. and Kragic, D. (2005). Receptive Field Cooccurrence Histograms for Object Detection. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS'05*, pages 84–89.

Ekvall, S., Kragic, D., and Jensfelt, P. (2007). Object Detection and Mapping for Service Robot Tasks. *Robotica*, **25**, 175–187.

Fairley, S., Reid, I., and Murray, D. (1998). Transfer of Fixation Using Affine Structure: Extending the Analysis to Stereo. *International Journal of Computer Vision*, **29**(1), 47–58.

Fischler, M. and Bolles, R. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, **24**(6), 381–395.

Frintrop, S. (2006). VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search. *Lecture Notes in Computer Science*, **3899**.

Geib, C., Mourao, K., Petrick, R., Pugeault, N., Steedman, M., Krüger, N., and Wörgötter, F. (2006). Object Action Complexes as an Interface for Planning and Robot Control. In *Workshop: Towards Cognitive Humanoid Robots at IEEE RAS Int Conf. Humanoid Robots*.

Gevers, T. and Smeulders, A. (1999). Color Based Object Recognition. *Pattern Recognition*, **32**(3), 453–464.

Grieg, D., Porteous, B., and Seheult, A. (1989). Exact Maximum A Posteriori Estimation for Binary Images. *Journal of Royal Statistical Society - B*, **51**(2), 271–279.

Harris, C. and Stephens, M. (1988). A Combined Corner and Edge Detector. In *Proceedings of the of the 4th Alvey Vision Conference*.

Hartley, R. and Zisserman, A., editors (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ.

Heidemann, G., Rae, R., Bekel, H., Bax, I., and Ritter, H. (2004). Integrating Context-Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference. *Machine Vision and Applications*, **16**(1), 64–73.

Hu, Y., Xie, X., Ma, W.-Y., Chia, L.-T., and Rajan, D. (2004). Salient Region Detection using Weighted Feature Maps based on the Human Visual Attention Model. In *IEEE Pacific-Rim Conference on Multimedia*, pages 993–1000.

Huebner, K., Björkman, M., Rasolzadeh, B., Schmidt, M., and Kragic, D. (2008a). Integration of Visual and Shape Attributes for Object Action Complexes. In *6th International Conference on Computer Vision Systems*, volume 5008 of *Lecture Notes in Artificial Intelligence*, pages 13–22. Springer-Verlag.

Huebner, K., Ruthotto, S., and Kragic, D. (2008b). Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping. In *IEEE International Conference on Robotics and Automation*, pages 1628–1633.

Itti, L. (2000). *Models of Bottom-Up and Top-Down Visual Attention*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, USA.

Itti, L. and Koch, C. (2001). Computational Modeling of Visual Attention. *Nature Reviews Neuroscience*, **2**(3), 194–203.

Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(11), 1254–1259.

Kamon, I., Flash, T., and Edelman, S. (1998). Learning Visually Guided Grasping: A Test Case in Sensorimotor Learning. *IEEE Transactions on Systems, Man and Cybernetics*, **28**(3), 266–276.

Koch, C. and Ullman, S. (1985). Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Human Neurobiology*, **4**(4), 219–227.

Koike, T. and Saiki, J. (2002). Stochastic Guided Search Model for Search Asymmetries in Visual Search Tasks. In *2nd International Workshop on Biologically Motivated Computer Vision*, volume 2525 of *Lecture Notes in Computer Science*, pages 408–417.

Kraft, D., Baseski, E., Popovic, M., Krüger, N., Pugeault, N., Kragic, D., Kalkan, S., and Wörgötter, F. (2008). Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. *International Journal of Humanoid Robotics*, **5**, 247–265.

Kragic, D. and Kyrki, V. (2006). Initialization and System Modeling in 3-D Pose Tracking. In *In IEEE International Conference on Pattern Recognition 2006*, pages 643–646.

Kragic, D., Björkman, M., Christensen, H., and Eklundh, J.-O. (2005). Vision for Robotic Object Manipulation in Domestic Settings. *Robotics and Autonomous Systems*, **52**(1), 85–100.

Kuniyoshi, Y., Kita, N., Sugimoto, K., Nakamura, S., and Suehiro, T. (1995). A Foveated Wide Angle Lens for Active Vision. In *International Conference on Robotics and Automation (ICRA95)*, pages 2982–2988.

Lee, K., Buxton, H., and Feng, J. (2003). Selective Attention for Cueguided Search using a Spiking Neural Network. In *International Workshop on Attention and Performance in Computer Vision*, pages 55–62.

Li, Z. (2002). A Saliency Map in Primary Visual Cortex. *Trends in Cognitive Sciences*, **6**(1), 9–16.

Longuet-Higgins, H. (1980). The Interpretation of a Moving Retinal Image. In *Philosophical Trans. Royal Society of London, B-208*, pages 385–397.

Longuet-Higgins, H. (1981). A Computer Algorithm For Reconstructing a Scene From Two Projections. *Nature*, **293**, 133–135.

Lowe, D. (1999). Object Recognition From Local Scale-Invariant Features. In *IEEE International Conference on Computer Vision*, pages 1150–1157.

Morales, A., Recatalá, G., Sanz, P., and del Pobil, A. (2001). Heuristic Vision-Based Computation of Planar Antipodal Grasps on Unknown Objects. In *IEEE International Conference on Robotics and Automation*, pages 583–588.

Morales, A., Chinellato, E., Fagg, A., and del Pobil, A. (2004). Using Experience for Assessing Grasp Reliability. *International Journal of Humanoid Robotics*, **1**(4), 671–691.

Moren, J., Ude, A., Koene, A., and Cheng, G. (2008). Biologically-Based Top-Down Attention Modulation for Humanoid Interactions. *International Journal of Humanoid Robotics*, **5**(1), 3–24.

Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. (2003). Development of a High-speed Multifingered Hand System and Its Application to Catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2666–2671.

Navalpakkam, V. and Itti, L. (2003). Sharing Resources: Buy Attention, Get Recognition. In *International Workshop Attention and Performance in Computer Vision*.

Nickerson, S. B., Jasiobedzki, P., Wilkes, D., Jenkin, M., Milios, E., Tsotsos, J., Jepson, A., and Bains, O. N. (1998). The ARK Project: Autonomous Mobile Robots for Known Industrial Environments. *Robotics and Autonomous Systems*, **25**(1–2), 83–104.

Oliva, A., Torralba, A., Castelhano, M., and Henderson, J. (2003). Top-Down Control of Visual Attention in Object Detection. In *International Conference on Image Processing*, pages 253–256.

Olshausen, B., Anderson, C., and van Essen, D. (1993). A Neurobiological Model of Visual Attention and Invariant Pattern Recognition based on Dynamic Routing of Information. *Journal of Neuroscience*, **13**(11), 4700–4719.

Paulus, D., Ahrlichs, U., Heigl, B., Denzler, J., Hornegger, J., Zobel, M., and Niemann, H. (1999). Active Knowledge-Based Scene Analysis. *Proceedings of the First International Conference on Computer Vision Systems*, **1542**, 180–199.

Petersson, L., Jensfelt, P., Tell, D., Strandberg, M., Kragic, D., and Christensen, H. I. (2002). Systems Integration for Real-World Manipulation Tasks. In *IEEE International Conference on Robotics and Automation, ICRA'02*, volume 3, pages 2500–2505.

Platt Jr., R., Fagg, A., and Gruppen, R. (2002). Nullspace Composition of Control Laws for Grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1717–1723, Lausanne, Switzerland.

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in Context. In *International Conference on Computer Vision*, pages 1–8.

Ramström, O. and Christensen, H. (2004). Object Detection using Background Context. In *Proc. International Conference of Pattern Recognition*, pages 45–48.

Rasolzadeh, B. (2006). KTH Attention-Table Dataset. In *http://www.e.kth.se/ babak2/database.htm*.

Rasolzadeh, B., Björkman, M., and Eklundh, J. (2006). An Attentional System Combining Top-Down and Bottom-Up Influences. In *Proc. International Cognitive Vision Workshop (ICVW06)*.

Rasolzadeh, B., Targhi, A. T., and Eklundh, J.-O. (2007). An Attentional System Combining Top-Down and Bottom-Up Influences. In *WAPCV*, pages 123–140.

Riddoch, M., Humphreys, G., Edwards, S., Baker, T., and Wilson, K. (2001). Seeing the Action: Neuriopsychological Evidence for Action-Based Effects on Object Selection. In *Nature Neuroscience, 4*, pages 84–88.

Sandini, G. and Tagliasco, V. (1980). An Anthropomorphic Retina-like Structure for Scene Analysis. *Computer Graphics and Image Processing*, **14**(3), 365–372.

Scassellati, B. (1998). A Binocular, Foveated, Active Vision System. Technical report, MIT AI Memo 1628.

Shimoga, K. (1996). Robot Grasp Synthesis: A Survey. *International Journal of Robotics Research*, **3**(15), 230–266.

Shiu, Y. and Ahmad, S. (1989). Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form Ax = Xb. *IEEE Transactions on Robotics & Automation*, **5**(1), 16–29.

Siagian, C. and Itti, L. (2007). Biologically-Inspired Robotics Vision Monte-Carlo Localization in the Outdoor Environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1723–1730.

Sigala, N. and Logothetis, N. (2002). Visual Categorization Shapes Feature Selectivity in the Primate Temporal Cortex. *Nature*, **415**, 318–320.

Sloman, A. (2001). Evolvable Biologically Plausible Visual Architectures. In *British Machine Vision Conference, BMVC'01*, pages 313–322.

Strat, T. and Fischler, M. (1989). Context-Based Vision: Recognition of Natural Scenes. In *23rd Asilomar Conference on Signals, Systems & Computers*, pages 532–536.

Strat, T. and Fischler, M. (1995). The Use of Context in Vision. In *IEEE Workshop on Context-Based Vision*.

Topp, E. A., Kragic, D., Jensfelt, P., and Christensen, H. I. (2004). An Interactive Interface for Service Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 3469–3475, New Orleans.

Treisman, A. and Gelade, G. (1980). A Feature Integration Theory of Attention. *Cognitive Psychology*, **12**, 97–136.

Tsai, R. and Lenz, R. (1988). Real Time Versatile Robotics Hand/Eye Calibration Using 3D Machine Vision. In *IEEE International Conference on Robotics and Automation*, pages 554–561.

Tsotsos, J. (1987). Analyzing Vision at the Complexity Level: Constraints on an Architecture, An Explanation for Visual Search Performance, and Computational Justification for Attentive Processes. Technical report.

Tsotsos, J. K., Culhane, S. M., Winky, W. Y. K., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling Visual Attention via Selective Tuning. *Artificial Intelligence*, **78**(1-2), 507–545.

Ude, A., Gaskett, C., and Cheng, G. (2006). Foveated Vision Systems with Two Cameras per Eye. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3457–3462.

Vijayakumar, S., Conradt, J., Shibata, T., and Schaal, S. (2001). Overt Visual Attention for a Humanoid Robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2332–2337.

Wörgötter, F., Agostini, A., Krüger, N., Shylo, N., and Porr, B. (2009). Cognitive Agents – a Procedural Perspective Relying on the Predictability of Object-Action-Complexes. *Robotics and Autonomous Systems*, **57**(4), 420–432.

Ye, Y. and Tsotsos, J. (1999). Sensor Planning in 3D Object Search. *Computer Vision and Image Understanding*, **73**(2), 145–168.

List of Figures

List of Footnotes

[1]The center-surround-differences are a computational model of the center-surround receptive fields composed by ganglion cells in the retina. For details on the across-scale subtraction we refer to Itti's original work.

[2]Note that dilation has been applied for the reprojected segments for the later application of point-based object hypotheses verification. The dilation, the grid approach, as also noisy and incomplete data from stereo cause that reprojections are often little larger or not completely covering the bodies.

[3]The notion of shape is here simplified into 3D size, meaning the approximate width, breadth and height of the object as listed in the intrinsic attribute list.

Active 3D scene segmentation and detection of unknown objects

Mårten Björkman and Danica Kragic

***Abstract*— We present an active vision system for segmentation of visual scenes based on integration of several cues. The system serves as a visual front end for generation of object hypotheses for new, previously unseen objects in natural scenes. The system combines a set of foveal and peripheral cameras where, through a stereo based fixation process, object hypotheses are generated. In addition to considering the segmentation process in 3D, the main contribution of the paper is integration of different cues in a temporal framework and improvement of initial hypotheses over time.**

I. Introduction

The next important milestone for embodied machine vision systems is to make them flexible and robust in a variety of environments and tasks. Recent examples of machine vision systems for humanoid robots [1] demonstrate the necessity for active aspects of the system, both in terms of actively changing the parameters of the vision system and interacting with the environment. Visual attention serves as a core process for generating hypotheses about the structure of the scene and allows the system to deal with the complexity of natural scenes. The requirements on machine vision systems are highly dependent on the task, and have historically been developed with this in mind. To deal with the complexity of the environment, prior task and context information have commonly been integrated with low level processing structures, the former being denoted as top-down and latter bottom-up principle. This has many times been motivated by human visual processing. Humans build a representation of a visual scene using a temporal process of integration of several scene 'glances', [2]. A cumulative memory allows them to detect and recall objects seen during several short, separate presentations even when these are several minutes apart. Likewise, in machine vision systems, generating hypotheses about objects in the scene is a necessary prerequisite for interaction. Although generation of hypotheses may be solved through a classical process of object recognition, our main interest is to generate hypotheses of *previously unseen* objects. This process may also help the recognition and classification processes by reducing the search space.

The main contribution of the work presented here is 3D scene segmentation based on the integration of several visual cues. However, this work should not be viewed as a typical work on image segmentation, since the hypotheses of objects are generated in 3D, thus facilitating shape attribution and pose estimation. We also show how segmentation can evolve over time and gradually produces better hypotheses. This is another important difference from the classical segmentation approaches that are typically demonstrated on a single image. We also evaluate the presented method using an active humanoid head in realistic scenarios. As said, this work relates to classical approaches to segmentation, however, most of these have been demonstrated only in the image space. Segmentation in 3D offers not only the possibility to attribute 3D regions based on their shape properties, [3], but also gives direct input to an object grasping and manipulation system, [4].

The work presented here is related to image segmentation methods such as GrabCut, [5] in that it models segmentation as a hypotheses generation and verification process. However, in the GrabCut approach only two hypotheses are used: one for the foreground and one for the background. We will show that in a 3D segmentation process, additional hypotheses increase the quality of the results. In addition, we employ belief propagation for verification of hypotheses, that differs from the energy minimization approaches of [5] and [6]. The most important difference and also a contribution is that our method uses a temporal framework and verifies the hypotheses over time, whereas methods of [5] and [6] work on a single image.

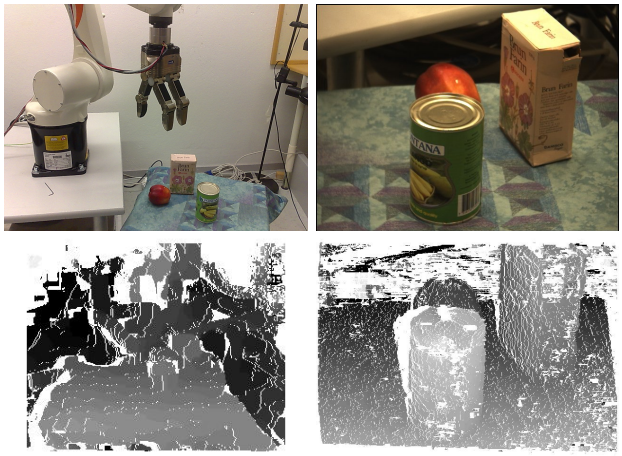


Fig. 1. Left: A peripheral view of a typical experimental scene (upper), with a corresponding disparity map (lower). Right: A foveal view of the same scene (upper) with a disparity map (lower).

The goal behind the presented work is to enable a vision guided robotic system to learn about its environment through interaction with the objects therein. First, the hypotheses of possible scene objects need to be generated within reasonable time. This means that an attention system that directs the vision system towards the most conspicuous parts of the

This work was supported by EU through the project PACO-PLUS, IST-FP6-IP-027657, and GRASP, IST-FP7-IP-215821 and Swedish Foundation for Strategic Research. The authors are with the Centre for Autonomous Systems and Computational Vision and active Perception Lab, CSC-KTH, Stockholm, Sweden. celle,dani@kth.se

scene is needed. Second, extraction of attributes related to an observed object often requires it first to be segregated from its background. With the attention system already presented elsewhere, [4] here we concentrate on the second problem, figure-ground segmentation of objects in typical indoor scenes.

A. Experimental platform

Our experimental platform includes the 7-joint Armar III robotic head, [7]. The stereo head carries four Point Grey Dragonfly cameras grouped in two pairs, a peripheral and a foveal one, see Fig. 1. These are parts of an existing vision system [4] that uses attention in the peripheral view to direct cameras towards nearby regions of interest. After gaze direction such regions are placed in fixation in the foveal view. Binocular disparities are exploited in both views, for gaze control in the peripheral view and for object analysis and manipulation in the foveal view.

Visual attention, gaze control and manipulation are beyond the scope of this paper, yet they serve as the context in which the presented segmentation approach is to be used. The disparity maps shown in Fig. 1 are computed using Stable Matching [8], a method that is able to cope with wide disparity ranges. The range we typically use for the foveal views, 64 pixels, is more than what most disparity methods are able to handle within reasonable time. Stable Matching is suitable for our needs, since instead of aiming for the highest possible density, it tries to minimize the number of false positive matches.

B. Assumptions

In typical indoor environments most physical objects are placed on flat surfaces. However, based on our previous work [9], an object may be impossible to separate from the surface: they may be similar in appearance[1]. In this paper we thus expand a typical framework for figure-ground segmentation with an additional model, that of a flat surface. A foreground object is defined as the object fixated on by the stereo system. Thus it is expected to be placed in the center of view at about zero disparity. In GrabCut [5], a foreground object is similarly defined by a given bounding box. We also assume that models change only slightly while the object is in fixation and that the system knows when the gaze is shifted and segmentation has to be reinitialized. Finally, the system should be able to operate autonomously through sequences of gaze shifts and tolerate disparity data that arises through non-perfect calibration and limited disparity search ranges.

II. Prerequisites

The segmentation method presented in this paper is based on measurements of colors and binocular disparities. Given these measurements the scene is divided into 3 parts; a foreground object, a flat surface and a background. We later describe a scheme with which model parameters can be estimated and images segmented on a per-pixel basis.

[1]See http://www.csc.kth.se/∼danik/HeadArmDemo-centering.avi for an example of using the system for object grasping.

A. Measurements and model parameters

An image, here assumed to be part of a stereo pair, contains image points that are characterized by their positions (x_i, y_i) and measured colors $c_i = (h_i, s_i, v_i)$ given in HSV space, with h_i being the hue, s_i the saturation and v_i the luminance value. Also associated to each such point is a measured binocular disparity d_i, that can either be a value within a given disparity range or be undefined. There are primarily two reasons for the disparity to be undefined; either a point lacks sufficient texture to be matched in stereo or it is occluded in one of the two images. We denote the total set of image measurements by $\mathbf{m} = \{m_i\}$, with each point characterized by $m_i = (p_i, c_i)$, where $p_i = (x_i, y_i, d_i)$ are the three spatial measurements and c_i is the color.

We assume each image point to originate from one of three possible scene parts; a foreground object $\mathbf{F}$, a planar surface $\mathbf{S}$ and a background $\mathbf{B}$, each of which a characterized by a corresponding model. The foreground $\mathbf{F}$ is assumed to be a connected set of 3D points representing some physical object in the center of the image and close to the fixation point. It is further assumed that the scene contains a large planar surface $\mathbf{S}$, upon which objects could be placed. The background $\mathbf{B}$ is defined as all points that neither belong to the foreground nor the planar surface. The scene part that a particular point p_i belongs to is given by a label $l_i \in L$, where $L = \{l_f, l_s, l_b\}$ is the set of values that corresponds to each scene part respectively.

The three different parts of the scene are modeled by a set of parameters $\theta = \theta_f \cup \theta_s \cup \theta_b$. These will be defined later in Section II-B. Given the measurements $\mathbf{m}$ our goal is to find the most likely parameter set θ and distribution of labels $\mathbf{l} = \{l_i\}$. The joint probability of $\mathbf{m}$ and $\mathbf{l}$ given θ can be written as

$$p(\mathbf{m}, \mathbf{l}|\theta) = p(\mathbf{m}|\mathbf{l}, \theta)p(\mathbf{l}|\theta) \tag{1}$$

with the measurement distribution given by

$$p(\mathbf{m}|\mathbf{l}, \theta) = \prod_i p(m_i|\theta_f)^{I_i^f} p(m_i|\theta_b)^{I_i^b} p(m_i|\theta_s)^{I_i^s} \tag{2}$$

and the prior label probabilities

$$p(\mathbf{l}|\theta) = \prod_k p(l_k) \prod_i \prod_{j \in N_i} p(l_i, l_j). \tag{3}$$

In the equations above, I_i^x equals 1 if $l_i = l_x$ and 0 otherwise, and N_i is the set of neighbors to point i. The priors in (3) will be defined later in Section III-A.

B. Scene part models

For all three scene parts we model the distributions of image point positions, disparities and colors. The spatial distributions of the background and surface parts are assumed to be uniform across the image space $\mathbf{X}$, i.e. $p(x_i, y_i|\theta_b) = p(x_i, y_i|\theta_s) = 1/N$, where $N = |\mathbf{X}|$ is the number of image points. Their counterparts in disparity space are modeled as Gaussians with $p(d_i|\theta_b) = n(d_i; d_b, \Delta_b)$ and $p(d_i|\theta_s) = n(d_i; \alpha_s x_i + \beta_s y_i + \delta_s, \Delta_s)$, where $d_s = (\alpha_s, \beta_s, \delta_s)$ are disparity parameters that belong to the surface model. Here

we denote by $n(x; \bar{x}, \Delta)$ a Gaussian distribution of a d-dimensional variable x, with mean $\bar{x}$ and covariance Δ,

$$n(x; \bar{x}, \Delta) = \frac{1}{\sqrt{(2\pi)^d |\Delta|}} \exp^{-\frac{1}{2}(x-\bar{x})^\top \Delta^{-1}(x-\bar{x})}$$

While the conditional probability of the background is the same for all image points, it varies for the flat surface. Note that $d = \alpha_s x + \beta_s y + \delta_s$ represents a plane in (x, y, d) space that, assuming a projective camera, corresponds to a plane also in the 3D metric space. The spatial positions of the foreground object are modeled using a single 3D Gaussian that includes both image point positions and disparities, with conditional probabilities given by $p(x_i, y_i, d_i|\theta_f) = n(p_i; p_f, \Delta_f)$. The disparity dimension is ignored for points with undefined disparities and for these points Δ_f is replaced by its projection in (x, y)-space.

The distributions of colors within a given scene part are assumed to be the same for all image points. We represent such distributions as 2D histograms, based on hue and saturation; $p(h_i, s_i|\theta_b) = H_b(h_i, s_i)$, $p(h_i, s_i|\theta_s) = H_s(h_i, s_i)$ and $p(h_i, s_i|\theta_f) = H_f(h_i, s_i)$. With color histograms included in the set of model parameters, the complete set is given by

$$\begin{aligned}\theta_f &= \{p_f, \Delta_f, c_f\},\\ \theta_b &= \{d_b, \Delta_b, c_b\},\\ \theta_s &= \{d_s, \Delta_s, c_s\},\end{aligned}$$

where c_f, c_b and c_s denote the color histogram bins stacked into vectors. The other parameters are the means and variances of the Gaussians mentioned above. The joint measurement conditionals can finally be summarized as

$$\begin{aligned}p(m_i|\theta_f) &= n(p_i; p_f, \Delta_f)\, H_f(h_i, s_i),\\ p(m_i|\theta_b) &= N^{-1} n(d_i; d_b, \Delta_b)\, H_b(h_i, s_i),\\ p(m_i|\theta_s) &= N^{-1} n(d_i; \alpha_s x_i + \beta_s y_i + \delta_s, \Delta_s)\, H_s(h_i, s_i).\end{aligned}$$

III. Estimating the model parameters

One way of estimating the model parameters θ would be to determine a maximum likelihood estimate for $p(\mathbf{m}|\theta)$ using the Expectation-Maximization (EM) algorithm, with all labels $\mathbf{l}$ treated as hidden variables. Given $p(\mathbf{m}, \mathbf{l}|\theta)$, that was defined in (1), the hidden variables can be eliminated through marginalization,

$$p(\mathbf{m}|\theta) = \sum_{\mathbf{l}} p(\mathbf{m}, \mathbf{l}|\theta).$$

The EM algorithm is based on maximization of an objective function $Q(\theta|\theta')$ that given a previous estimate θ' is guaranteed to increase $p(\mathbf{m}|\theta)$. In the first step of the algorithm, the Expectation step, $Q(\theta|\theta')$ is expressed as the expected value of $\log p(\mathbf{m}, \mathbf{l}|\theta)$ with respect to the conditional distribution $w(\mathbf{l}) = p(\mathbf{l}|\mathbf{m}, \theta')$ under the previous estimate θ', that is

$$Q(\theta|\theta') = \sum_{\mathbf{l}} w(\mathbf{l}) \log p(\mathbf{m}, \mathbf{l}|\theta). \tag{4}$$

The model parameters θ are updated in the second step, the Maximization step, through maximization of $Q(\theta|\theta')$. This two-step procedure is then repeated until convergence.

As can be seen in (4), the algorithm essentially performs a summation over the conditional distribution $w(\mathbf{l})$. Unfortunately, this fact makes the EM algorithm intractable for our purpose. In our case labels from neighboring image points are assumed to be dependent. This means that the summation has to be done across all 3^N possible combinations of labels, where N is the number of image points, rather than $3N$ combinations that would otherwise have been the case.

To make summation computationally tractable, we introduce an approximation that treats labels as if they are in fact independent. We do this by replacing the conditional distribution $w(\mathbf{l})$ with the product of the marginal distributions for each unobserved label, that is

$$\hat{w}(\mathbf{l}) = \prod_i w(l_i) = \prod_i p(l_i|\mathbf{m}, \theta').$$

Since a measurement m_i at a given point only depends on the label l_i at that point, not on neighboring labels, the summation in (4) becomes

$$Q_1(\theta|\theta') = \sum_i \sum_{l_i \in L} w(l_i) \log p(m_i, l_i|\theta). \tag{5}$$

With dependencies ignored the joint probability for a single point (see (1) and (2)) can be written as

$$p(m_i, l_i|\theta) = p(m_i|l_i, \theta) p(l_i),$$

where

$$p(m_i|l_i, \theta) = p(m_i|\theta_f)^{I_i^f} p(m_i|\theta_b)^{I_i^b} p(m_i|\theta_s)^{I_i^s}.$$

Note that it is only when marginal distributions are summed up to produce an estimate of θ that dependencies between labels are ignored. The marginals $w(l_i)$ themselves determine the final segmentation and are computed with dependencies taken into consideration.

A. An iterative two-stage approach

Our optimization approach consists of two stages, that are iterated until either convergence or the number of iterations reaches a given maximum. Given an initial estimation of the conditional marginals for all individual labels, or the marginals from the previous iteration, the model parameters are estimated by maximizing $Q_1(\theta|\theta')$ in (5), where θ' are the parameters from which the marginals were computed. The corresponding update functions for all foreground parameters can be found in the appendix.

In the second stage the conditional marginals $w(l_i) = p(l_i|\mathbf{m}, \theta)$ are recomputed for each label. This is done using loopy belief propagation [10]. First, however, we have to rewrite the equations into energy functions suitable for belief propagation. From Bayes' rule and using the fact that m_i only depends on l_i, we have that

$$p(\mathbf{l}|\mathbf{m}, \theta) = \frac{p(\mathbf{m}|\mathbf{l}, \theta) p(\mathbf{l}|\theta)}{p(\mathbf{m}|\theta)} = \frac{\prod_i p(m_i|l_i, \theta)}{\prod_i p(m_i|\theta)} p(\mathbf{l}|\theta)$$

and from the label priors in (3)

$$p(\mathbf{l}|\mathbf{m}, \theta) = \frac{\prod_k p(m_k|l_k, \theta) p(l_k)}{\prod_k \sum_{l \in L} p(m_k|l_k = l, \theta)} \cdot \prod_i \prod_{j \in N_i} p(l_i, l_j).$$

The network of image points can be considered a Markov Random Field (MRF), with the first factor in the equation above representing cliques of one point each and the second involving pairs of points. The corresponding energy functions are given by the negative logarithms of these factors. Note that the second factor represents a smoothing term that is intended to capture the spatial continuity in typical scenes, and penalizes solutions that include discontinuities.

With no penalty if two neighboring points are labeled the same and a constant penalty when labeled differently, the joint probabilities of two neighboring points can be modeled using the Potts model [11], [12]

$$p(l_i, l_j) = \exp^{-V_{i,j}[l_i \neq l_j]}$$

where $[C]$ denotes an indicator function that takes a value 1 if C is true and 0 otherwise. Similar to [13] and [5] we use a pair-wise penalty based on the difference in luminance between image points;

$$V_{i,j} = 50 \, \exp^{-\beta(v_i - v_j)^2},$$

where

$$\beta = (2\langle (v_i - v_j)^2 \rangle)^{-1}.$$

and $\langle \cdot \rangle$ denotes the expectation over an image.

An alternative solution to the problem above could have been based on maximum a posteri (MAP) estimates, instead of the conditional marginals of each label. A local maximum of $p(\mathbf{m}, \mathbf{l}|\theta)$ is searched, while alternating between keeping $\mathbf{l}$ or θ fixed. This is what is done in GrabCut [5]. It is known that if there are only two possible labels per point, an exact MAP solution can be found using graph-cuts [14], and even if the problem becomes NP-hard with more than two labels, there are efficient approximate solutions at hand [6]. While the EM algorithm estimates model parameters by an enumeration over all possible configuration of labels, a MAP based approach would use only one such configuration.

Since we have an interest in the model parameters themselves, in particular those of the foreground, a MAP approach can become problematic. What frequently occurs in figure-ground segmentation are cases where the interpretation of a particular non-textured background region alternates between foreground and background. This leads to model parameters radically change from frame to frame. EM takes such uncertainties into consideration and their respective probabilities are weighted in when parameters are estimated.

B. Initialization

The iterative scheme described above is initialized through a rough segmentation of the image into the three scene parts, using the assumptions mentioned in Section I-B. At this stage only pixels for which disparities exist are considered. Occluded or non-textured areas are ignored until after initialization. From the assumption that the foreground object is in fixation, image points located within a 3D ball are sought and assigned to the foreground model $\mathbf{F}$. The size of the ball is set so that its projective size is equals to half the image height.

Among the remaining image points a flat surface is sought using random sampling with 1000 trials. For each such trial three points are randomly selected and the parameters of a plane $d = \alpha_s x + \beta_s y + \delta_s$ are determined. Since the robot head knows its approximate orientation, planes that are not horizontal enough can immediately be discarded. Among the non-discarded planes, the plane with the highest number of matching image points across the whole image is then selected. A point is considered as matching if its disparity is within 2 pixel values from that of the plane. Points that match the selected plane equation are finally assigned to the surface model $\mathbf{S}$, while the rest are assigned to the background $\mathbf{B}$. Once image points have been assigned, the iterative scheme in section III-A can get started.

IV. Adding dependency over time

In an active vision system image point positions, disparities and colors can be expected to change only slightly from one frame to the next, at least as long as there are no rapid gaze shifts. This consistency over time can be exploited in the estimation of model parameters. In our system we do this by regarding the estimated parameters from the previous frame, θ^t, as measurements when considering the current. Instead of searching the maximum likelihood estimate for $p(\mathbf{m}|\theta)$, we do it for $p(\mathbf{m}, \theta^t|\theta)$.

With labels and point measurements independent of θ^t, the objective function $Q_1(\theta|\theta')$ in (5) is replaced by

$$Q_2(\theta|\theta') = \sum_i \sum_{l_i \in L} w(l_i) \log p(m_i, l_i|\theta) + \log p(\theta^t|\theta) \quad (6)$$

The transition probabilities $p(\theta^t|\theta)$ have three factors, one for each scene part, that is

$$p(\theta^t|\theta) = p(\theta_f^t|\theta_f) p(\theta_b^t|\theta_b) p(\theta_s^t|\theta_s),$$

where

$$\begin{aligned} p(\theta_f^t|\theta_f) &= n(p_f^t; p_f, \Lambda_f)\, n(c_f^t; c_f, \sigma_c^2 I)\, g(\Delta_f^t; \Delta_f, S_f), \\ p(\theta_b^t|\theta_b) &= n(d_b^t; d_b, \Lambda_b)\, n(c_b^t; c_b, \sigma_c^2 I)\, g(\Delta_b^t; \Delta_b, S_b), \\ p(\theta_s^t|\theta_s) &= n(d_s^t; d_s, \Lambda_s)\, n(c_s^t; c_s, \sigma_c^2 I)\, g(\Delta_s^t; \Delta_s, S_s). \end{aligned} \quad (7)$$

Here Λ_f is the expected variance over time for the positional parameters of the foreground, while Λ_b and Λ_s are corresponding variances for the disparity parameters of the background and surface models. The expected variance of the color histogram bins is denoted σ_c^2. The remaining functions $g(\Delta^t; \Delta, S)$ capture the assumed consistency of covariance matrices over time and are defined as follows.

A. Time consistency of covariance matrices

Assume we would like to estimate a covariance matrix Δ given some measurements $\{x_i\}$, and a previously estimated covariance matrix Δ^t at time t. If we assume the underlying distribution changes gradually from one instance in time to the next, we need some way to express its consistency over

Fig. 2. Segmentation results for every fourth frame of a sequence generated by the attention system. Segmentation is re-initiated after each saccade.

Fig. 3. Segmentation results for various scenes. The 9th frame in a sequence is shown in each case.

Fig. 4. Segmentation results with foreground, surface and background models. The images show the 1st, 3rd, 5th and 7th frames of a sequence.

time. In this study we assume the consistency between Δ and Δ^t to be given by

$$g(\Delta^t; \Delta, S) = \left(\frac{1}{2\pi|\Delta|}\right)^{S/2} \exp\left(-\frac{S}{2}\sum_i \lambda_i \mu_i^\top \Delta^{-1} \mu_i\right),$$

where μ_i and λ_i are the eigenvectors and eigenvalues of Δ^t, and S is the strength of the dependency. The equation can be interpreted as $\prod_j p(y_j|\Delta^t)$, where S samples $\{y_j\}$ are drawn from a Gaussian distribution with zero mean and variance Δ^t. If we assume there are no measurements $\{x_i\}$ at time t and Δ only depends on Δ^t, then an estimate Δ^* can be determined from $\arg\max_\Delta g(\Delta^t; \Delta, S)$. We first compute the logarithm of the consistency function

$$\log g(\Delta^t; \Delta, S) = -\frac{S}{2}(\log(2\pi|\Delta|) - \sum_i \lambda_i \mu_i^\top \Delta^{-1} \mu_i,$$

and its derivative with respect to Δ^{-1}

$$\frac{\delta}{\delta \Delta^{-1}} \log g(\Delta^t; \Delta, S) = \frac{S}{2}\left(\Delta - \sum_i \lambda_i \mu_i \mu_i^\top\right).$$

Setting the derivative to 0 results in

$$\Delta^* = \sum_i \lambda_i \mu_i \mu_i^\top = \Delta^t.$$

Hence, if there are no measurements, then Δ will be directly given by Δ^t. In this case the consistency strength factor S has no influence on the result. It will become important, however, when consistency over time is combined with the image point measurements.

V. Experimental evaluation

We performed a series of realistic experiments with objects scattered on a table. A short sequence[2] of foveal views from such an experiment can be seen in Fig. 2. This sequence illustrates how the system is able to rapidly segment an object in its foveated view. For each view the attention system has controlled the cameras and placed an object hypothesis in the center of view.

Using a typical Core 2 processor, the segmentation, including disparity extraction, requires about a second per update

[2]Available as a movie at http://www.csc.kth.se/∼danik/ICRA2010_AVI.avi

with 640×480 pixel images and five iterations per update. For all these experiments we set the expected variances over time of the position parameters (defined in (7)) to Λ_f = diag{1000, 1000, 4}, $\Lambda_b = 25$ and Λ_s = diag{0.0001, 0.0004, 1}. We used normalized color histograms with 10×10 bins each, with an expected variance of $\sigma_c^2 = 0.00001$ for each bin. The time consistency values for the covariance matrices were set to $S_f = S_b = S_s = N$, i.e. the number of image points. Finally, the prior label probabilities were assumed to be $p(l_f) = 20\%$, $p(l_b) = 40\%$ and $p(l_s) = 40\%$. All remaining model parameters were estimated from image and disparity measurement, using the procedure described in Section III.

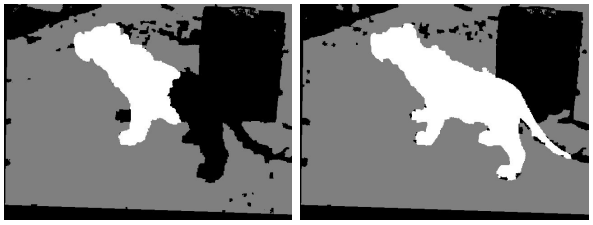


Fig. 5. Point labels of the first and last images of Fig. 3. Pixels labeled as surface points are shown in gray, while white pixels indicate foreground.

Fig. 6. Segmentation results without an obvious surface plane. The lower images show pixels labeled as surface points in gray.

Fig. 7. Segmentation results without a surface model. The images show the 1st and 7th frames of a sequence.

A. Segmentation results

Using the above mentioned method, segmentation results can be seen in Fig. 3 for a selection of scenes, some more challenging than others. Since the inner part of the cup in the third image lacks reliable disparities and its shade resembles that of a background object, a fragment is still labeled as background after the 9th update. The last image shows an case where the assumption that the foreground object can be described as an ellipsoid fails. The tail of the giraffe will eventually be included, but never the legs. Fig. 4 shows how segmentation evolves over time. With the initial assumption that the foreground can be represented by a ball around zero disparity, it takes a few updates for the model to extend to include the whole cat. Labeling results for the first and last updates can be seen in Fig. 5. As shown by the gray pixels, the table top is captured by the surface model already from the first update.

We also consider how the method behaves if no distinct flat surface exists in the scene. Two such examples are shown in Fig. 6. From the gray pixels we observe that the background and surface models have essentially changed order, while the foreground segmentation is unaffected. The surface model finds some non-physical plane across the background objects. The thickness of the plane is gradually extended to include large parts of the scene. The background model is unable to compete, since image points are assumed to be uniformly distributed, even though scene points are typically not.

Fig. 8. Segmentation results without disparity measurements. The images show the 1st and 7th frames of a sequence.

Fig. 9. Segmentation results without color measurements. The images show the 1st and 7th frames of a sequence.

B. Benefits of multiple cues and models

The method presented here differs from the traditional figure-ground segmentation: it exploits multiple cues for segmentation (colors, positions and disparities) and together with the foreground and background hypotheses it also includes a third, that of a flat surface. Fig. 7-9 show how important these additions are by showing what happens when they are removed. If no flat surface hypothesis were added, one would get results similar to those of Fig. 7. Since the initial ball around the cat includes parts of the table and these parts are located on about the same depth, the foreground segment cannot differentiate between cat and table. The

foreground segment will grow from frame to frame and eventually the whole table will be included.

The behavior could become even worse when disparity measurements are not taken into consideration. Fig. 8 shows an example of that. Without disparities the surface model loses its function and becomes just another background model. Cues that would otherwise have prevented the table top from being included in the foreground become even weaker. Similar behaviors can sometimes be observed in GrabCut, [5], when the initial selected region contains too much of a similarly colored background. Samples from such a false background may result in a distinct peak in the foreground color histogram, which strengthens the hypothesis that these samples do in fact belong to the foreground in next update. With high-quality disparities and a flat surface hypothesis, segmentation often becomes trivial, even without color measurements. However, for regions with unreliable or undefined disparities, color measurements can still be beneficial, as can be seen in Fig. 9.

VI. Discussion and Conclusions

Generating hypotheses about objects in natural scene is a prerequisite for enabling robots to interact with the environment. In this paper, we have presented an active vision system consisting of a two sets of stereo cameras: one for foveal and one for peripheral vision. The system is used for 3D segmentation of visual scenes based on integration of several cues. The main application of the system is to serve as a visual front end and generate object hypotheses for objects not known *a-priori*. The active part of the system is the use of a stereo based fixation process, where objects hypotheses are generated and improved over time. The main contributions of the work is i) that the process of segmentation is considered in 3D thus also providing the input for direct interaction with the environment; ii) the process of temporal segmentation is modeled, showing how the quality of object hypotheses improves over time.

Experimental evaluation demonstrates segmentation of objects in natural scenes with some of the underlying assumptions being violated. Still, the presented method performs well and provides several good object hypotheses. We believe that this is an important result towards equipping robots with the capability of detecting novel objects in the environments and use metric information for direct grasping and manipulation of objects. Our current work explores the use of the system for generation of 3D shape attributes of objects. In addition, we will extend the method for automatic 3D object model generation using several different views of the same object and thus improve the quality of generated grasps.

Appendix

For conciseness we denote the foreground marginal probability of point i by $w_f^i = w(l_i{=}l_f)$. With the color histogram bin corresponding to the same point denoted by b_i, the value of this bin is $c_{f,b_i} = H_f(h_i, s_i)$, where c_f is the foreground color histogram vector. Given the objective function

$$Q_f(\theta) = \sum_i w_f^i \log p(m_i, l_f|\theta_f) + \log p(\theta_f^t|\theta_f)$$

the following update functions of the foreground model can be derived:

$$\frac{\delta Q_f(\theta)}{\delta p_f} = \sum_i w_f^i \Delta_f^{-1}(p_f - p_i) + \Lambda_f^{-1}(p_f - p_f^t) = 0 \Rightarrow$$

$$p_f \leftarrow (\sum_i w_f^i + \Delta_f \Lambda_f^{-1})^{-1}(\sum_i w_f^i p_i + \Delta_f \Lambda_f^{-1} p_f^t)$$

$$\frac{\delta Q_f(\theta)}{\delta \Delta_f^{-1}} = \frac{1}{2}(\sum_i w_f^i \Delta_f + S_f(\Delta_f - \Delta_f^t) - \sum_i w_f^i (p_f - p_i)(p_f - p_i)^\top) = 0 \Rightarrow$$

$$\Delta_f \leftarrow \frac{\sum_i w_f^i (p_f - p_i)(p_f - p_i)^\top + S_f \Delta_f^t}{\sum_i w_f^i + S_f}$$

$$\frac{\delta Q_f(\theta)}{\delta c_{f,j}} = \frac{1}{c_{f,j}} \sum_{b_i = j} w_f^i + \frac{1}{\sigma_c^2}(c_{f,j}^t - c_{f,j}) = 0 \Rightarrow$$

$$c_{f,j} \leftarrow \hat{c}_{f,j} / \sum_i \hat{c}_{f,i}, \text{ where}$$

$$\hat{c}_{f,j} = \frac{1}{2} c_{f,j}^t + \frac{1}{2} \sqrt{{c_{f,j}^t}^2 + 4\,\sigma_c^2 \sum_{b_i = j} w_f^i}$$

Update functions for the background and surface models can be derived similarly.

References

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

[2] D. Melcher, "Persistance of visual memory for scenes," *Nature*, vol. 412, no. 6845, p. 401, 2001.

[3] K. Huebner, M. Björkman, B. Rasolzadeh, M. Schmidt, and D. Kragic, "Integration of Visual and Shape Attributes for Object Action Complexes," in *6th International Conference on Computer Vision Systems*, ser. LNAI, vol. 5008. Springer-Verlag, 2008, pp. 13–22.

[4] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, "An Active Vision System for Detecting, Fixating and Manipulating Objects in Real World," *International Journal of Robotics Research*, 2009, to appear, available from http://ijr.sagepub.com/pap.dtl.

[5] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut – interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[7] T. Asfour, K. Regenstein, P. Azad, J. Schröder, and R. Dillmann, "Armar-iii: A humanoid platform for perception-action integration," in *Proceedings of the International Workshop on Human-Centered Robotic Systems (HCRS)*, 2006.

[8] R. Sara, "Finding the largest unambiguous component of stereo matching," in *Proceedings 7th European Conference on Computer Vision (ECCV)*, vol. 2, May 2002, pp. 900–914.

[9] D. Kragic, M. Bjorkman, H. I. Christensen, and J.-O. Eklundh, "Vision for robotic object manipulation is domestic settings," *Robotics and Autonomous Systems*, pp. 85–100, Jun 2005.

[10] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, pp. 1–41, 2000.

[11] R. Potts, "Some generalized order-disorder transformation," *Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 106–109, 1952.

[12] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 609–628, July 1990.

[13] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *Proceedings of International Conference on Computer Vision (ICCV)*, vol. I, 2001, pp. 105–112.

[14] D. Greig, B. Porteous, and A. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989.

Spatio-temporal modeling of grasping actions

Javier Romero, Thomas Feix, Hedvig Kjellström, Danica Kragic

Abstract— **Understanding the spatial dimensionality and temporal context of human hand actions can be used to provide representations for programming grasping actions in robots, and use these for designing new robotic and prosthetic hands. Natural human hand motion is highly non-linear and of high dimensionality. However, for specific activities such as handling and grasping of objects, it has been proposed that the commonly observed hand motions lie on a lower-dimensional non-linear manifold in hand posture space. This is also true for human motion in general. Although full body human motion is well studied within Computer Vision and Biomechanics, there is very little work on the analysis of hand motion. In this paper we use Gaussian Process Latent Variable Models (GPLVMs) to model the lower dimensional manifold of human hand motions, during object grasping in particular. We show how the technique can be used to embed high-dimensional grasping actions in a lower-dimensional space suitable for modeling, recognition and mapping. The technique is evaluated both on synthetic and real data.**

I. Introduction

Modeling of human hand motion is receiving an increased interest in areas such as computer vision, graphics, robotics and psychology [1]. The goal of the work presented here is to study and model the spatial dimensionality and temporal context of human hand actions to i) provide representations for programming grasping actions in robots, and ii) use these for designing new robotic and prosthetic hands. In robotics, it has been argued that continuous motion mapping from human to robot requires suitable spatio-temporal representations of human and robot motions, [3]. However, most of the work on grasp mapping is based on hand-designed grasping taxonomies considering a *discrete* hand postures, [2].

The main contribution of the work presented here is the study of spatial and temporal context of human grasping actions. We develop a low-dimensional grasping hand motion model that also allows to evaluate state-of-the-art grasping taxonomies. We go beyond single, discrete hand postures by considering the temporal aspects of grasping actions. Gaussian Process Latent Variable Model (GPLVM) is used to model the manifolds of high-dimensional human grasping motions. The adopted technique has been used in several recent studies of human body motion but has not been used to study human hand motion. The main motivation is to allow for generation of grasping actions in lower-dimensional spaces when, for example, task constraints need to be taken into account, [3]. When the dimensionality of the generation space is lower, control strategies applied to the robot hand become simpler. We also compare the developed methodology with several classical dimensionality reduction techniques.

Despite the wide use of subspace representations in human body motion analysis [5], [6], the work on human hand motion in general is very limited. An analysis of low-dimensional embeddings of human hand grasping was performed in [7]. However, the data was recorded from subjects imagining grasp actions instead of applying them. Furthermore, the low-dimensional space was created with PCA, which is limited by its linear nature [8]. The benefits of non-linear dimensionality reduction schemes is shown in [9], where a 2D space is used for the control of robotic grasps. The GPLVM method adopted here places a Gaussian Process (GP) prior over the generative mapping from latent space to a data space. Through marginalization of this mapping, the marginal likelihood of observed data given the latent locations can be found. The latent locations are then found by maximizing this likelihood. Due to the flexibility of GPs, the generative mapping is not constrained to be linear as is the case when using PCA or PPCA. Moreover, it has been shown that it is more efficient than other techniques like Isomap when dealing with noisy and incomplete training data, [10].

This paper is structured as follows. In Section II the related work is presented. Section III gives a brief introduction to GPLVM. Section IV and V present the evaluations on synthetic and real data. Finally, the work is summarized in Section VI.

II. Contributions and Related Work

Human grasping research in robotics concentrates on the design of grasping taxonomies based on the observation of human grasping actions. Grasp types or hand postures are purely based on the author's intuition and some specific needs of the field the taxonomy is used in. In general, there is very little consensus between the different taxonomies. Our recent study, [2] analyzes several taxonomies proposed in the areas of robotics, biomechanics or medicine. An important observation is that the taxonomies have not been contrasted against the actual data extracted from subjects performing the grasps. Work in that direction was presented in [7]. Subjects were asked to shape the hand as if they were grasping different objects. A CyberGlove was used to record 15 joint angles of the grasping hand. This data was projected onto a low-dimensional space with PCA. The main conclusion was that the first component of the projected data accounts for 80% of the variance of the data.

In our work, we further develop these ideas in several directions. First, we consider the whole grasping sequence instead of just a single grasp posture. This facilitates the spatial and temporal reconstruction of a grasping action. Second, the latent space is reconstructed from end-effector

data (fingertip position and orientation relative to the palm) instead of joint angles. Thus, we avoid the problem of proximal joints having a higher impact on the position of the fingertip. The end-effector data is also easier to translate to other embodiments than the joint angle data. Third, instead of studying how different objects are grasped, we study how different grasps are performed. The motivation for this is that some objects can be grasped in different ways depending on the goal of the grasp (pick a pen or write with a pen). Fourth, due to the non-linearity of the human hand motion, we use non-linear methods to construct the low-dimensional representation space.

The work by Ciocarlie et al. [8], [11] focuses on reducing the complexity of robotic grasping through the use of PCA. The low-dimensional space extracted in [7] is used both for reducing the complexity of grasp space exploration [8] and for mapping between an operator and a simulated hand [11]. Since the space contains only hand postures where the final grasp has been achieved, the approach phase of the grasp is not taken into account. In [9] data from a Vicon optical motion capture system is used to create a latent space for "Interactive Control of a Robot Hand" using Isomap. The data is a concatenation of different grasps and tapping demonstrations. Contrary to our approach, the authors do not provide any study of the similarity between the demonstrated grasps. In Section V-C, we will also discuss the performance of Isomap for our purposes.

In the context of full-body human motion, GPLVMs have been employed both for visual tracking of full-body motion [5], [6] and for classification of full-body actions [12]. Modeling the dynamics in embedded spaces of lower dimensionality decreases the amount of training data needed [5] and facilitates the generation of natural and physically plausible motion [13]. To model the pose and motion of the human jointly, [5], [6] use Gaussian Process Dynamical Models (GPDMs), an extension of GPLVMs with a latent dynamical model. In this extension, the optimization of the latent variables includes the probability of the temporal sequence of latent variables, modeled as a Gaussian Process whose parameters are marginalized.

In this paper, we use GPLVMs for creating a low-dimensional grasp space in which we can reason about the similarities and differences between a set of predefined grasps [2]. This lower dimensional space is optimized for minimizing the reconstruction error from it to the observed space. An immediate application of this latent space is a non-parametric dynamic model of grasping actions for tracking and classification; however, this is out of the scope of the work presented here. For our purpose, we do not model dynamics explicitly as in [5], [6], but include back-constraints (Section III) that indirectly enforce temporal continuity in the latent space. This avoids the unimodal nature of the GPDM dynamics. The created GPLVM model allows the generation of concatenated grasping actions with natural transitions. This can be done by applying constraints in the latent space in a similar way as constraints are applied in [4].

III. Theoretical formulation

Let D denote the dimension of the data space and q the dimension of the latent space. Given N observations, the matrix containing the data points is denoted $Y \in \mathbb{R}^{N\times D}$ and the matrix of the corresponding points in the latent space is $X \in \mathbb{R}^{N\times q}$. The marginal likelihood P of the datapoints, given the latent positions and the parameters θ, is a product of D independent Gaussian processes [14]:

$$P(Y|X,\theta) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{N}{2}}|K|^{\frac{1}{2}}} e^{-\frac{1}{2}y_j^T K^{-1} y_j} \qquad (1)$$

where $y_j \in \mathbb{R}^{N\times 1}$ is the jth column of the data matrix and $K \in \mathbb{R}^{N\times N}$ is the covariance matrix. To obtain the latent representation of Y one has to maximize Eq. 1 wrt X and θ. In general, this optimization has many solutions since the function is not convex [14]. To make the optimization tractable, GPLVM sets a prior over the possible mappings given by θ and optimizes the latent space points X.

A. Covariance Functions

The covariance matrix K in Equation 1 is determined by the covariance or kernel function k:

$$K_{i,j} = k(x_i, x_j) \qquad (2)$$

The choice of the covariance function is critical, since it defines the behaviour of the solution. This is also an advantage of the method, since it allows adaptation to the specific needs of the task and the dataset at hand. The kernel function needs to generate a valid covariance matrix, i.e. a positive semidefinite kernel matrix. Therefore, the class of valid kernels is the same as the class of Mercer functions. For practical purposes it should also be possible to calculate the gradient of the kernel with respect to the latent space, since gradient based optimization is used to calculate the maxima of Equation 1. A special case is the linear kernel

$$k(x_i, x_j) = \alpha x_i x_j \qquad (3)$$

since the solution of the optimization is then identical to the PCA solution [14]. Most commonly, the covariance matrix is determined by a sum of several different kernels, like the Radial Basis Function (RBF), bias and noise kernels. A popular covariance function is the RBF kernel:

$$k(x_i, x_j) = \alpha\ e^{-\frac{\gamma}{2}(x_i - x_j)^T (x_i - x_j)} \qquad (4)$$

where α defines the output variance and the inverse kernel width γ controls the smoothness of the function. By using a smooth covariance function like the RBF kernel, we encode a preference towards smooth generative mappings in the GP prior. This implies that points close in the latent space will remain close in the observed space (when projected using the mean prediction of the GP). However, it is not guaranteed that the inverse is true, i.e. points close in the observed space remain close in the latent space. This is further discussed in the next subsection. In addition to the RBF kernel we also include a bias term which accounts for translations in the data and a white noise term.

B. Back Constraints

As stated above, a GPLVM in its basic form does not guarantee that a smooth inverse exists to the generative mapping [15]. However, this can be incorporated into the model by representing the latent locations x_i in terms of a smooth parametric mapping g_j from the observed data y_i.

$$x_{ij} = g_j(y_i, a) = \sum_{n=1}^{N} a_{jn} k_{bc}(y_i - y_n) \tag{5}$$

where k_{bc} is the back constraint kernel. This means that the maximum likelihood solution of these parameters a rather than the latent locations are sought. This is referred to as a back-constrained GPLVM. In addition to constraining the latent location to preserve the local smoothness of the observed data, previously unseen data can be projected onto the latent space in an efficient manner by pushing them through this back-mapping.

IV. Evaluation on Synthetic Data

The proposed technique was first evaluated on synthetic data. For this purpose, Poser 7, a commercial human modeling software, was used to model 31 grasp types as static hand postures. The choice of those grasp postures is motivated in Section II. We excluded "Distal Type" and "Tripod Variation" due to their very specific nature.

The transportation component of a grasp movement varies significantly depending on the orientation and distance of the object to the hand. Therefore, hand pose is here defined as the pose of the fingers relative to the palm. For simplicity, we used the pose parameters of Poser, where each finger was assigned five DoF; three for the angles of the proximal finger joints and one for each of the two distal finger joint angles. Thus, the full hand pose is defined by 25 parameters. For each grasp, the end posture was defined by setting the parameters manually. Three intermediate poses were then generated by linearly interpolating between the end and a standard starting posture shared by all the grasps. Linear interpolation was chosen because of its simplicity, despite of not completely resembling human way of grasping. This gave five samples per grasp, resulting in a total of 155 points in the data set.

A. Low dimensional representation

We created the GPLVM latent space spanned by this data. For this purpose, the Matlab FGPLVM toolbox [16], [15] was used. Several different configurations of the GPLVM parameters (with and without back constraints, different back constraint types, variation of parameters) were analyzed. **We used Scaled conjugate gradient optimization to obtain maxima of Equation 1**. The best results were achieved with a kernel composed of RBF, bias and noise, and kernel based regression back constraints with an RBF kernel. The back-constraint takes the form of a regression over a kernel induced feature space defined using a RBF kernel. The inverse width of the kernel is a free parameter and was set to 0.001 by inspection of the scatter response matrix of the training data. Following [7], [9], we selected a dimensionality of 2 for the latent space, simplifying the visualization of the results. The resulting latent space in Figure 1 has a very distinct star shape. This is due to the special nature of the data set, with a common starting posture and linear interpolation to the different end postures. In the middle of the star is the resting position of the hand. If one moves outside along a branch, a specific grasp type will be formed. This star shape is a property of the dataset, and can also be seen in subspaces found by PCA or Isomap [17]. Nevertheless the star shape is most pronounced with GPLVM.

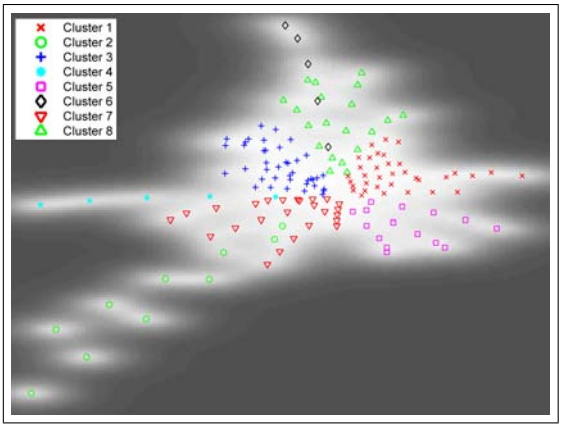


Fig. 1. Grasp space spanned by synthetic data of 31 grasp actions. See Figure 2 for the allocation of the grasp types.

B. Similarity Measure and Clustering of Grasps

The similarity between grasps was measured as the Earth Mover's Distance (EMD) between the two sets of points in latent space. EMD shows how costly is to convert one point set into another, coping well with the problem when variances of the two sets differ substantially. It also shows better robustness wrt outlier as, for example, the Hausdorff distance.

Thw similarity between grasps can be visualized by clustering them into grasp groups. The clustering algorithm chosen is average linkage clustering algorithm also known as UPGMA, from the Matlab Statistical Toolbox. As opposed to some popular clustering techniques like kmeans, UPGMA does not require the metric to be Euclidean.

The minimum number of clusters which properly subdivides the central region of the space was 8. For a number of clusters below 8 significantly different grasps in the center of the latent space were assigned to a big centered cluster. The resulting clusters are shown in Figure 2.

As an example, all grasps in *cluster seven* are precision grasps ([2], [18]) with middle, ring and little finger extended. *Cluster one* contains grasps that are mostly three fingered grasps and the position of the thumb is abducted. The dataset created with Poser has also provided a number of insights about the general principles of grasp clustering, but the latent space shape and the clusters are heavily affected by the limitations of the dataset itself. Therefore, in the next section we present results on real human grasps.

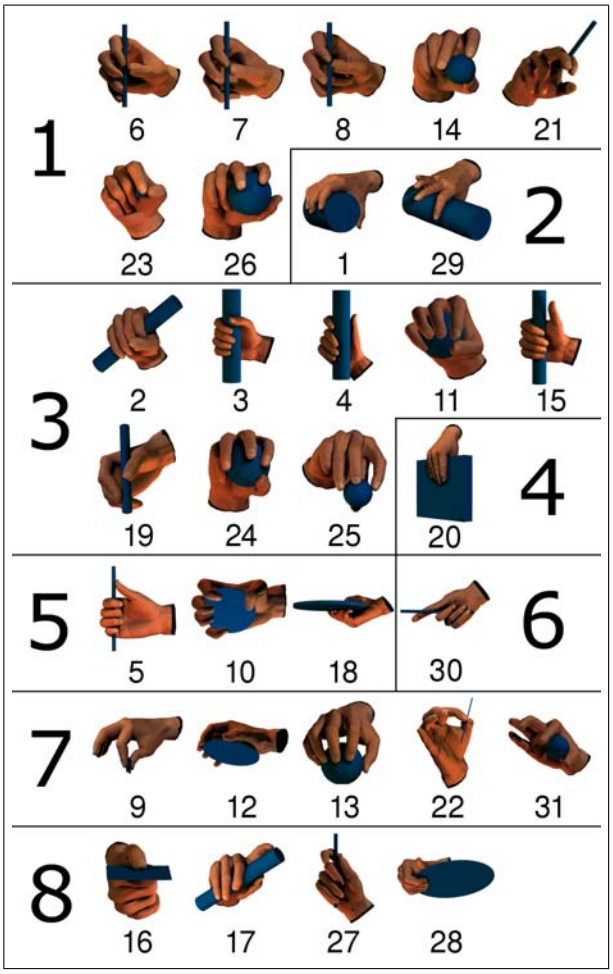


Fig. 2. Clusters of the grasp types for the synthetic data. The number of clusters for the algorithm was manually set to 8.

V. Evaluation on Real Data

The data was generated with 5 subjects (3 male, 2 female). All subjects are right handed and have not reported any hand disabilities. The average hand length is 185,2 $\pm$ 13,3mm and hand width is 81,1 $\pm$ 7,4mm. A Polhemus Liberty system with six magnetic sensors was used for recording the data. The spatial and angular resolution of each sensor is 0.8 mm and 0.15 degrees respectively. One sensor was applied to each fingertip, positioned on the fingernail and one was placed on the dorsum of the hand. See Figure 3 for an image of the markers applied to the hand. The subjects were asked to perform the same 31 grasp types as used in the synthetic data set. They were shown a picture of each grasp and a demonstration of the grasp was performed for the most difficult ones. The data was then further processed as follows:

1) Calibration that aligns the coordinate systems of the sensors with the actual anatomical direction.
2) Transformation of the fingertip data into the wrist coordinate system. As discussed in Section IV, the global movement of the hand depends strongly on the distance and orientation to the object. To provide some invariance to these aspects, the hand pose is defined in terms of the relative position and orientation of the fingertip sensors with respect to the wrist.
3) Translation of the position of the fingertip origin to the center of the distal finger segment and normalization of the dimensions to a standard range.

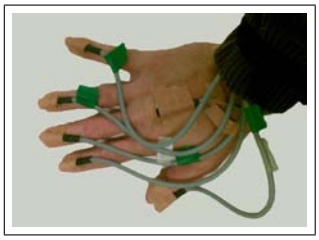


Fig. 3. Placement of the sensors. Five sensors are placed on the fingertips and one is positioned on the wrist.

The sensors create a space of dimensionality 35 where each of the 5 sensors has 7 dimensions: 3 for position and 4 for orientation (we used quaternions to represent rotations). As data for the dimensionality reduction algorithms, we used the constraint experiments with the 31 grasp types. From each trial we took 30 equally distributed samples creating a constant length. Overall this resulted in a data matrix of size 4650×35. This space is over determined, since the human hand has only around 25 DoF. Despite having higher dimensionality, working directly with the sensor data avoids the problem of different importance of joint angles in a kinematic chain [19] and obviates the complex problem of inverse kinematics in human hands.

A. Low dimensional representation of the grasp movements

We trained a GPLVM with the 31 grasps executed from five different subjects as described in the previous section. We introduced RBF back constraints to the same GPLVM as described in Section IV. The inverse width parameter was set to 0.001 by inspection. The model was initialized with different dimensionality reduction methods (PPCA, Isomap, LLE) and the one with lowest reconstruction error was kept. In our case this was an initialization with PPCA and the result of the optimization can be seen in Figure 4. **Thomas, check if the next paragraph makes sense. Iberall should have explained this in "The grasping hand"** We can observe that, although the space has a common starting point in the lower right corner, the shape of the low dimensional space is not "star-like" as opposed to the synthetic space. The main reason for this discrepancy is the difference in the starting position. The synthetic dataset started with the hand in the position, which the hand adopts when totally relaxed (all fingers slightly flexed). In contrast the real experiments had the "flat hand" as start posture, where all fingers are extended. In the beginning of the movement the subject flexes the fingers since all grasp types involves flexed fingers to a certain degree. This common movement forces that all trajectories move along the same direction. Additionally the assumption of a linear movement between starting and end posture is not supported by the data.

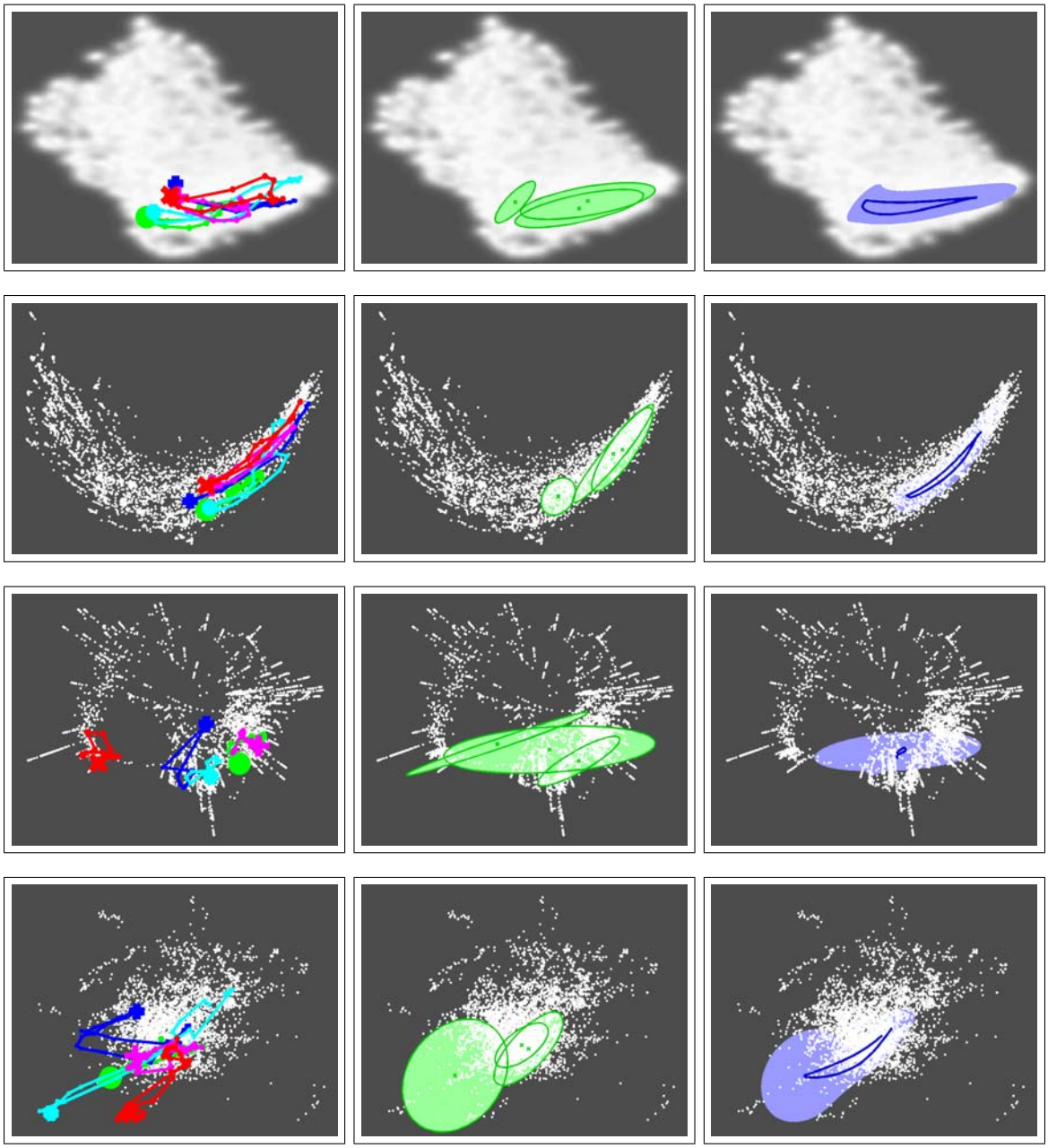


Fig. 5. From top to bottom: GPLVM, PCA, ISOMAP, LLE. From left to right: projection of grasp number 1 into latent space, GMM fitting, GMR regression. The other grasp types show similar patterns.

B. Gaussian Mixture Regression of Grasps

As opposed to the synthetic data, the real data contains multiple subject demonstrations. Therefore, the representation of each grasp in latent space should encompass temporal information (so that it is not just a point as in [7]) as well as multiple subject variance. We have used Gaussian Mixture Regression (GMR) [20], [4] for representing each grasp. We will briefly introduce this representation. More information can be found in [20], [4]. First the datapoints in latent space (bidimensional data, see first column of Figure 5) are extended with a time dimension. Then this data (three dimensions) is fitted into a Gaussian Mixture Model (GMM)(second column of Figure 5) by an expectation-maximization procedure initialized with K-means. Empirically, we found that using more than 3 gaussians did not improve the quality of the fitting. Based on that mixture of gaussians a hand posture is inferred for each time step by using GMR. This creates a continuous path through the latent space that describes the grasp (third column of Figure 5). That path has a mean and a variance. The paths corresponding to each of the 31 grasps can be found in Figure 6. The GMM/GMR representation of the grasps is a powerful tool that can be used for several purposes. One is the generation of new actions under some constraints [4]. In our case, this could help to generate an action composed

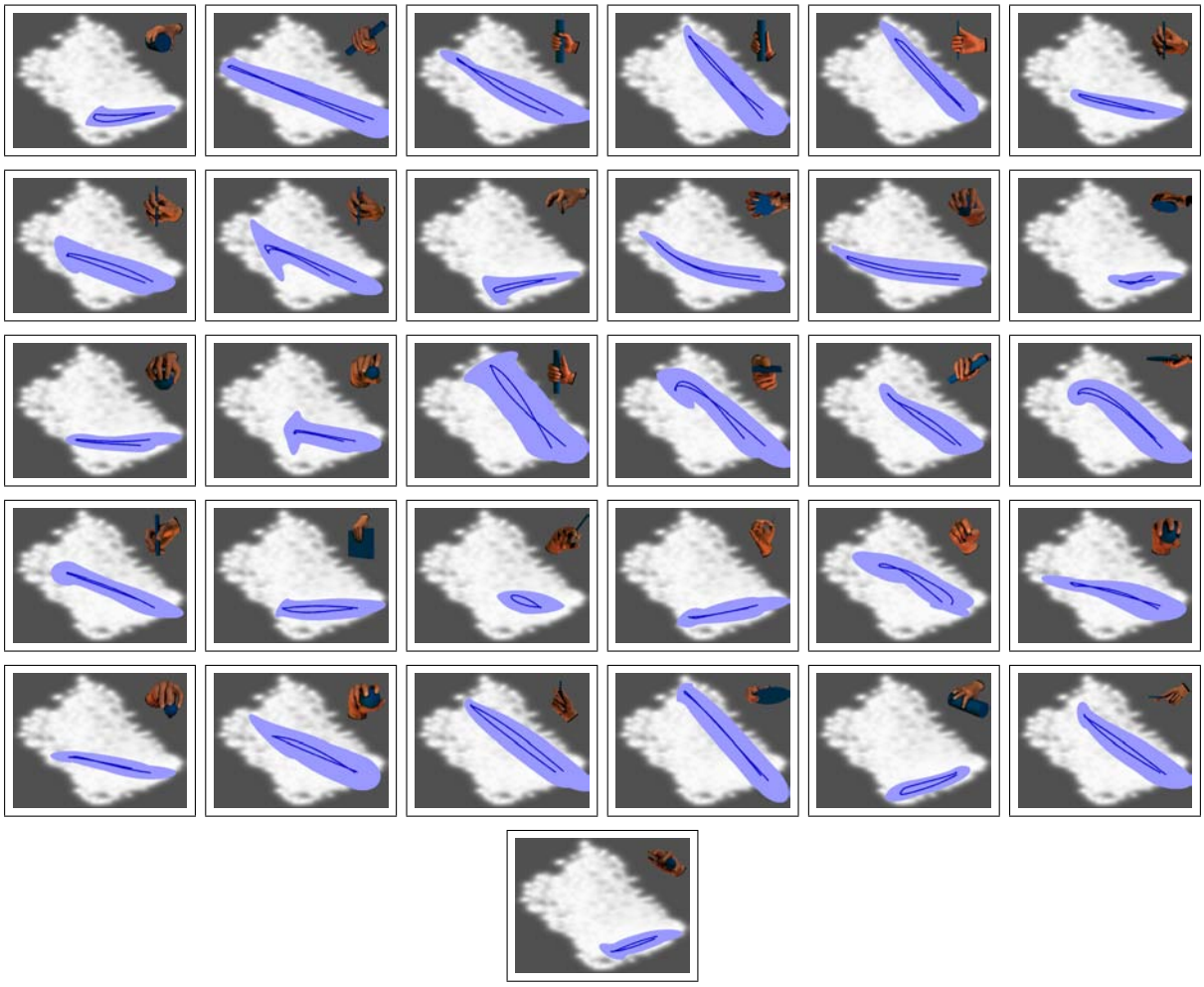


Fig. 6. GMR regression on the 31 grasp movements of all subjects. The dark line indicates the mean trajectory and the light area correspond to the uncertainty. The grasp are sorted, so the first row contains grasps 1 to 7 and so on.

of two grasps without coming back to the rest position between them. The second grasp can be constrained to start in a specific pose or after a specific time frame of the first grasp. The GMR can be optimized taking into account that constraint, providing in that way a smooth transition between those grasps.

C. Comparison of Dimensionality Reduction Algorithms

For comparison, other dimensionality reduction algorithms were applied to the same dataset of real human grasps. Again the latent space dimension was set to be 2. Algorithms used were Principal Components Analysis (PCA), Isomap (Matlab algorithm from [17]) and Locally Linear Embedding (LLE) (Matlab code from [21]). Figure 5 shows the low dimensional trajectories of all subjects performing grasp 1 and as background the corresponding latent space. This grasp is a typical example and the other grasp types show a similar pattern throughout all dimensionality reduction algorithms. The points of the PCA solution lie on an "arc" and the starting position is on the right side. This shape seems to be due to PCA being a linear method. It can only unravel the global motion in the data. As the hand moves to grasp the object, it advances in leftward direction along this arc. Since this arc is rather narrow there is little distinction between different grasp trajectories and fine details of the manifold cannot be extracted.

Isomap shows some sort of star-like structure, but one branch does not represent one grasp type as would be expected. Also the ability to generalize between subjects is not present, the trajectory of each subject is different without showing common trends. Increasing or decreasing the numbers of neighbors did not improve the result, so either the neighborhood size is too small or the locally linear assumption is violated.

LLE fails to discover any meaningful structure. All datapoints are centered in a certain location without any inner structure or common trajectories for grasp types.

Compared to the latent space created with GPLVM (Figure 4) those algorithms are not capable of separating the grasps that well and still preserve continuity in the trajectories. PCA is limited since it is a linear method; Isomap and LEE fail since they are based on local distance measurements which were disrupted by noise. Of course these problems also alter the GMM/GMR algorithm, so that the output is

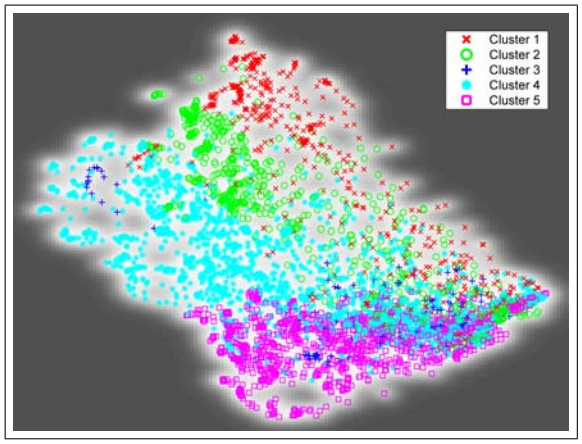


Fig. 4. Grasp space spanned by the execution 31 grasp types by five subjects. See Figure. 7 for a depiction of the grasps belonging to the clusters.

nearly a point (Isomap) or the trajectory has a very high variance (LLE). The ability to generalize between subjects is also visible in PCA, but the whole space is very packed and the trajectories of all grasp types are within a very small area. Only GPLVM is able to create a space where each grasp type has a distinct pattern similar for all subjects and yet uncover fine details of the low dimensional manifold.

D. Similarity Measure and Clustering of Grasps

We used GMM/GMR to measure similarity between human grasps. Since we have a probabilistic model for each grasp in the latent space (through their GMM representation), we can compute how likely it is that each point x in the space is generated by a grasp g_i.

$$p(x|g_i) = \sum_{k=1}^{3} \pi_k^{g_i} \mathcal{N}(x|\mu_k^{g_i} \sigma_k^{g_i})$$
$$p(g_j|g_i) = \prod_{\forall x \in g_i} p(x|g_i)$$
$$s(g_j, g_i) = (p(g_j|g_i) + p(g_i|g_j))/2$$

The product of the likelihoods of points in grasp g_j being generated by grasp g_i give us a measure of how well is g_j supported by the g_i model. Note that this measure is not symmetric. We can define the similarity between two grasps $s(g_j, g_i)$ as the average of those two quantities. We performed average linkage clustering (from the Matlab Statistical Toolbox, also known as UPGMA) based on this similarity measure. The result of the algorithm can be seen in Figure 7. Note that although synthetic hands are used for visualization purposes, the clusters were computed based on similarity between real human grasps. The number of clusters was chosen to be 5 since further subdividing the clusters overfits the data, i.e. *cluster four* was split into two groups with similar characteristics. Reducing the number of clusters resulted in large, too general clusters.

The grasps in *cluster one* resemble each other quite well. They all are power grasps with all four fingers in contact with the object. In addition the thumb is in a very adducted and extended position. The fingers are all in a very similar position, the MCP joint is rather extended, but the PIP and DIP joints are strongly flexed.

Cluster two is constructed by grasps that have a "straight" (extended MCP and IP joint) and mostly adducted thumb. Side opposition (see [18] for a description of the concept) is dominant in grasps 16, 27, 30 or at least there are some aspects that side opposition is involved as in grasps 17 and 18. None of those grasps is a precision grasp.

Cluster three is the only grasp in this cluster. This is due to the high variability of the grasp since fingertip positions are not restricted. Most subjects formed this grasp with all fingers extended, but one subject flexed the ring and the middle finger. Also the index and the middle finger, which are in contact with the object can be bent to a certain degree without affecting the stability of the grasp. Overall it seems that this grasp is formed in a rather extended position, this explains why the center of that grasp is very close to the starting position.

The biggest group of grasps is in *cluster four*. This group is quite diverse and it offers less distinct properties than the other groups. Yet all four fingers are all in a mid-flexed positions and the flexion increases towards the little finger. This is a clear difference to cluster five, where the little finger is in an extended position. In addition the thumb is mostly abducted, except grasp 23 where it is adducted.

Cluster five has a distinct inner structure. The horizontal direction in latent space modulates the overall extension/flexion of the fingers, whereas the vertical direction changes the individual index finger flexion.

In addition to those clusters properties, there are some general tendencies of the latent space. First, the further away a grasp is from the starting position (on the right side of the latent space) the more flexed the fingers will be. This is due to the fact that the starting position is with fingers and thumb totally extended and the transition between grasp types is smooth. The clusters seem to be elongated in the start-end posture direction. This makes sense, since the whole movement was taken into account when clustering the grasp types.

In the grasp taxonomy of [2] the thumb plays a crucial role in classifying the grasp types. The clusters, which were created here tend to go in accordance with this thumb classification, but there are some conflicts. This seems to be because the clustering algorithm gives each finger the same importance, where as in [2] the thumb plays a prominent role. Some grasp types do not employ all fingers, which means that potentially some finger positions are not relevant for the stability of the grasp. Currently those finger positions are taken into account with the same importance as fingers in contact with the object. Further work will be necessary to focus on fingers in contact with the object.

VI. Conclusion

The goal of the work presented here, differently from all the existing grasp taxonomies, was to model the spatial

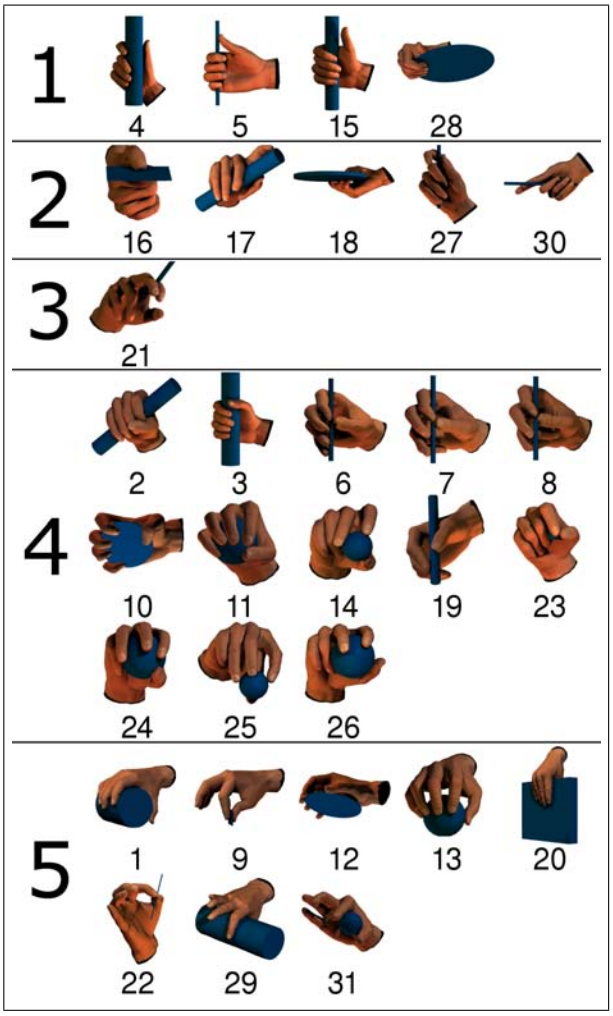


Fig. 7. Clusters of the human grasps. The number of clusters for the algorithm was manually set to be 5.

dimensionality and temporal context of hand actions. Instead of studying how different objects are grasped, we study how different grasps are performed. Apart from the important insights of human hand motion, the developed technique has also been used to evaluate the state-of-the-art taxonomies. We have shown how the technique can be used to embed high-dimensional grasping actions in a lower-dimensional space suitable for modeling, recognition and mapping. Considering the whole grasping sequence instead of just a single grasp posture facilitates the spatial and temporal reconstruction of a grasping action. The method is evaluated on both synthetic and real data.

An immediate application of the extracted latent space is a non-parametric dynamic model of grasping actions for tracking and classification. We do not model dynamics explicitly but include back-constraints that indirectly enforce temporal continuity in the latent space. This avoids the unimodal nature of the GPDM dynamics. The created GPLVM model potentially allows the generation of concatenated grasping actions with natural transitions. Thus, one idea is to apply constraints in the latent space in a similar way as in [4]. Together with the evaluation of the representation for classification this remains our future work.

REFERENCES

[1] A. Erol, G. N. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, pp. 52–73, 2007.

[2] T. Feix, R. Pawlik, H. Schmiedmayer, J. Romero, and D. Kragic, "A comprehensive grasp taxonomy," in *Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, Poster Presentation*, June 2009.

[3] S. Bitzer and S. Vijayakumar, "Latent spaces for dynamic movement primitives," in *Humanoids 2009*, 2009.

[4] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[5] R. Urtasun, D. J. Fleet, and P. Fua, "3D people tracking with gaussian process dynamical models," in *CVPR*, 2006, pp. I: 238–245.

[6] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[7] J. S. M. Santello, M. Flanders, "Postural hand synergies for tool use," in *The Journal of Neuroscience*, 1998.

[8] M. T. Ciocarlie, C. Goldfeder, and P. K.Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *IROS*. IEEE, 2007, pp. 3270–3275.

[9] A. Tsoli and O. C. Jenkins, "Neighborhood denoising for learning high-dimensional grasping manifolds," in *IROS*. IEEE, 2008, pp. 3680–3685.

[10] R. Urtasun, D. J. Fleet, A. Geiger, J. Popović, T. J. Darrell, and N. D. Lawrence, "Topologically-constrained latent variable models," in *ICML '08: Proceedings of the 25th international conference on Machine learning*, ser. ACM International Conference Proceeding Series, vol. 307. New York, NY, USA: ACM, 2008, pp. 1080–1087.

[11] M. T. Ciocarlie, S. T. Clanton, M. C. Spalding, and P. K. Allen, "Biomimetic grasp planning for cortical control of a robotic hand," in *IROS*. IEEE, 2008, pp. 2271–2276.

[12] R. Urtasun and T. Darrell, "Discriminative gaussian process latent variable model for classification," in *International Conference on Machine Learning*, 2007, pp. 927–934.

[13] S. Bitzer, S. Klanke, and S. Vijayakumar, "Does dimensionality reduction improve the quality of motion interpolation?" 2009.

[14] N. D. Lawrence, "The gaussian process latent variable model," The University of Sheffield, Department of Computer Science., Tech. Rep., 2006.

[15] N. D. Lawrence and J. Quinonero-Candela, "Local distance preservation in the gp-lvm through back constraints," in *ICML06*. New York, NY, USA: ACM, 2006, pp. 513–520.

[16] N. Lawrence and A. Hyvärinen, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[17] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, December 2000.

[18] T. Iberall, G. Bingham, and M. A. Arbib, "Opposition space as a structuring concept for the analysis of skilled hand movements," *Experimental Brain Research Series*, vol. 15, pp. 158–173, 1986.

[19] C. H. Ek, P. Torr, and N. Lawrence, "Gaussian process latent variable models for human pose estimation," in *Machine Learning for Multimodal Interaction*, 2008, pp. 132–143.

[20] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[21] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

Learning Task Constraints for Robot Grasping using Graphical Models

D. Song, K. Huebner, V. Kyrki and D. Kragic

***Abstract*— This paper studies the learning of task-relevant features that allow grasp generation in a goal-directed manner. We show how an object representation and a grasp generated on it can be integrated with the task requirements. The scientific problems tackled are (i) identification and modeling of such task constraints, and (ii) integration between a semantically expressed goal of a task and quantitative constraint functions defined in the continuous object-action domains. We first define constraint functions given a set of object and action attributes, and then model the relationships between object, action, constraint features and the task using Bayesian networks. The probabilistic framework deals with uncertainty, combines a-priori knowledge with observed data, and allows inference on target attributes given only partial observations. We present a system designed to structure data generation and constraint learning processes that is applicable to new tasks, embodiments and sensory data. The application of the task constraint model is demonstrated in a goal-directed imitation experiment.**

I. INTRODUCTION

A major challenge in robotics is the integration of symbolic task goals and low-level continuous representations. As an example, recent work in the area of path planning addresses the importance of modeling uncertainty in pose estimation and robot localization [1]. In the research area of object grasping and manipulation, the problem becomes a formidable challenge. Objects have many physical attributes that may constrain planning of a grasp, as also robots have limited sensorimotor capabilities due to their various embodiments.

Considering the problem at hand, multiple approaches take their inspiration from imitation studies in developmental psychology: infants are able to infer the intention of others from early on, and understand and reproduce the underlying task constraints through own actions [2]. More explicitely, this goal-directed imitative ability is obtained along multiple stages in a developmental roadmap, both through the infant's own motor *exploration* (trial and error) and through the *observation* of others interacting with the world (imitation learning), see reviews in [2], [3].

Inspired by these findings, roboticists follow a similar developmental approach in order to design architectures for artificial agents [3], [4], [5], [6]. Most of these works, however, focus on the exploratory stage, where robots obtain object affordances through their empirical interaction with the world. The affordances being modeled are measured as the salient changes in the agent's sensory channels, which are interpreted as effects of specific actions applied on objects [5]. As an example, an effect of poking a ball is making it roll. Though it is an important step for a robot to discover this motor ability, another necessary step to achieve goal-directed behavior is to link this immediate motor act and its effects (as to poke the ball and let it roll), to the conceptual goal of an assigned task (such as to provide the ball to a child). While trial-and-error-based exploration can be seen as inefficient to solve such goal learning problems, human supervision is helpful in this respect.

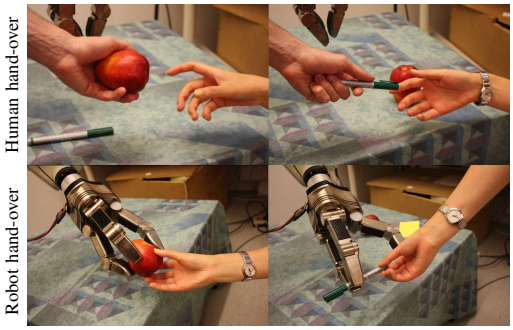


Fig. 1. The idea of goal-directed imitation and task constraint learning in a 'hand-over' task: though the embodiments, and thus hand configuration spaces, are very different, both hands follow similar task-based constraints.

This motivates an idea different from the classical developmental studies in path planning in such a way that it incorporates task-specific inputs from a human teacher. Thus, a system would be able to learn natural, goal-oriented types of grasps in a more efficient way. We clarify this idea in the hand-over task shown in Fig. 1. Such a task requires enough free area for another person to grasp the object. Thus, the robot should learn that an important constraint feature for this task is free area. There are numerous similar examples, e.g. pouring water from a cup requires the opening of a cup uncovered, and using a knife needs the robot to grasp the handle part. We believe these links can efficiently be learned by the proposed input from a human expert. In this work, we develop such a method for learning of task goals and task relevant representations. The learning is performed in a high-dimensional feature space that takes into account different object representations and robot embodiments together with an input from a teacher.

II. RELATED WORK

Deriving quantified constraints from conceptual task goals presents a challenge similar to integrating high-level rea-

D. Song, K. Huebner and D. Kragic are with KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, www: http://www.csc.kth.se/cvap, e-mail addresses: {dsong,khubner,danik}@kth.se.

V. Kyrki is with Lappeenranta University of Technology, Finland, Department of Information Technology, www: http://www.it.lut.fi, e-mail address: Ville.Kyrki@lut.fi.

soning with low-level path planning and control systems in robotics. The main challenges originate from the representational differences in the two research fields. A recent study [7] addresses this problem through statistical relational models. The authors use Bayesian logic networks to generate a knowledge base for a high-level symbolic reasoner, and integrate the inference results into a robot controller. Another work [8] proposes a coherent control, trajectory optimization, and action planning architecture. They apply the probabilistic inference-based methods and the dynamic Bayesian networks to integrate across all levels of representations.

Recently, imitation learning [9] and the concept of internal (forward and inverse) models [10] have received considerable attention in the field of robotics. The work described in [3], [4] implements an internal model through a probabilistic framework using Bayesian networks. This model formalizes the developmental imitation learning processes inspired by human infants [2]. In [11], it is shown that the internal models which represent the brain circuitry subserving sensorimotor control also participate in action recognition. They are used to predict the goal of observed behavior, and activate the correct actions to maintain or achieve the 'goal' state. A later work [12] extends the use of an internal model to the domain of visual-manual tasks. The authors implement a mental state inference function that can predict intention of using a hammer (nailing, prying or holding) based on how an actor is grasping it.

A very recent work closely related to ours is the affordance model by Montesano *et al*. [5]. The authors adopt a self-supervised, developmental approach where the robot first explores its sensory motor capabilities, and then interacts with objects to learn their affordances. A Bayesian network is used to capture the statistical dependencies between actions, object features and the observed effects of the actions. The authors demonstrate an application of the affordance model for a robot to perform goal-directed imitation.

Concluding, we observe that most of the named references are either considering higher-level planning systems [3], [7], or different domains with less complexity [8], [11], [12]. Even [5], though placed in the same domain of grasp affordance learning, is applied on fairly simple manipulation actions (tap or grasp) with discretely valued properties. Especially the latter describes a major drawback regarding the applicability in real world environments, which have to consider continuous and uncertain domains, be it in the acquisition of object features, path planning or motor control.

A. Motivation

In our work, we directly approach the task-oriented grasping problem considering characteristics of a real robot system. We facilitate the generation of according sensor and actor data, using a selected grasp planning system [13] in a grasp simulation environment [14]. On the one hand, a simulator allows to capture embodiment-specific motor capabilities (by using different hand models), and also to include the wrench-space based grasp quality measures in order to evaluate their relevance for certain task requirements. On the other hand, it enables supervised learning where knowledge of human experts can efficiently be used to bring the semantic task into the constraint learning loop. A concept showing evidence for the intuitivity and efficiency of incorporating task-constrained information through human tutoring, providing expertise about semantics, has been implemented in [15], [16]. Our grasps will be acknowledged by human experts to be suitable for a given manipulation task, to let the system learn the underlying structure of the feature space in a probabilistic framework.

To realize this, we take up the widely-used practice of applying a Bayesian Network (BN) [17]. In our case, this model will be used to (i) encode the statistical dependencies between object attributes, grasp actions and a set of task constraints; and to (ii) link the symbolic task goals to quantified constraints by exploiting the co-occurrence of the stimuli in different sensory channels, much alike to similar mechanisms in the human brain [18]. As such, our system models both categorical information (tasks) and the continuous action and object features through a mixed BN with continuous and discrete variables. In addition, we incorporate multiple object representations for similar object attributes. By applying Bayesian inference, the model can fill in missing object attributes, thus compensating for perceptual ambiguities caused by noise in the sensor data.

The main contributions of our work are (i) introducing a semi-automated method for acquiring manually annotated, task-related grasps; (ii) learning of probabilistic relationships between a multitude of task-, object- and action-related features with a mixed Bayesian network; (iii) thus acquiring a hand-specific concept of affordance, which maps symbolic representations of task requirements to the continuous constraints; additionally, using a probabilistic framework, we can easily extend the object and action spaces, and allow flexible learning of novel tasks and adaptation in uncertain environment. Finally, our model can be applied to a goal-directed imitation framework, which allows a robot to learn from humans despite differences in their embodiments.

III. DEFINITION OF FEATURE SUBSETS

To introduce our approach, we first identify four distinct subsets of features which play major roles in the consideration of a task-oriented grasp: task features, object features, action features, and constraint functions. These will define a frame for the creation of a Bayesion network learning approach which will be presented in Section IV. Using the definition of subsets, we can later flexibly instantiate a network with a specific constellation of network nodes, as will be demonstrated in Section V.

A. Task Features

In our notation, a *task* $T \in \mathcal{T} = \{t_1, ..., t_{n_T}\}$ refers to a 'basic task' that involves grasping or manipulation of a single object. According to the hierarchical task representation of [19], such a basic task is formally defined as a *manipulation segment* which starts and ends with both hands free and the object (or environment) at the stationary state. These

manipulation segments are the building blocks for a complex manipulation task. Though there may be an infinite number of complex tasks, we assume the basic building blocks form a finite set of object manipulation tasks. We further choose our task representation at the level of manipulation segments as each of them has an independent goal directly constraining how to grasp an object.

B. Object Features

We define an object feature set $\mathbf{O} = \{O_1, ..., O_{n_O}\}$ specifying the attributes (e.g. shape and size) and/or categorical (e.g. type or identity) information of an object. The features in $\mathbf{O}$ are not necessarily independent. The same attribute, such as shape, can be represented by different variables dependent on the capabilities of the perceptual system and the current object knowledge. For instance, eccentricity and convexity can be estimated from any kind of point cloud or mesh, while 3D shape representations like Zernike descriptors [20] can be used when a complete and dense 3D model of an object is available, i.e. when the object is known. Though apparently redundant, a system-dependent object representation offers flexibility in generalization across possibly different vision systems which can provide various levels of object knowledge.

C. Action Features

We define an action feature set $\mathbf{A} = \{A_1, ..., A_{n_A}\}$ describing the static, object-centered, *kinematic* grasp features, which may be the direct outputs of a grasp planner. The action feature set $\mathbf{A}$ may include properties like the pre-shape configuration, e.g. in terms of joint value vector; or a categorical pre-shape notion according to a taxonomy [21], like cylindrical grasp; or a latent space representation, as those of Eigengrasps [22]. In the same way, the final grasp configuration, the hand position and orientation, or a representation of tactile feedback can be represented in $\mathbf{A}$.

D. Constraint Functions

Finally, we let $\mathbf{C} = \{C_1, ..., C_{n_C}\}$ define a set of *constraint functions*; we term these to be a range of variables representing functions of both object and action features. Each constraint is therefore clearly dependent on and links between certain subsets of $\mathbf{O}$ and $\mathbf{A}$. As an example in a grasp scenario (like in Fig. 1), one may define the enclosure of the center-of-mass as a binary constraint feature, which obviously depends on both the specific object and action features; in our example: the center-of-mass and the pose and configuration of the hand. Thus, constraint features form the basic elements of general, task-dependent constraints in the sense that they can be used to quantitatively interpret the 'goal' or the 'requirements' of a given task.

We emphasize that this idea is an important aspect of our overall motivation: if one can identify certain constraint functions to be fundamental for a specific task, they must hold information which is independent of a specific object, or a specific hand. In our example, if enclosing the center-of-mass of an object is necessary for a task, this should hold for all objects and all hands.

IV. BAYESIAN NETWORK MODELING AND LEARNING

Given a complementary set of variables $\{T, \mathbf{O}, \mathbf{A}, \mathbf{C}\} = \mathbf{X}$, our focus is to model the dependencies between their elements, particularly those involving the task constraints $\mathbf{C}$. We model these dependencies through a Bayesian network (BN) [17]. A BN encodes the relations between the set of random variables $\mathbf{X} = \{X_i\}_{i=1,...,n}$. Each node in the network represents one variable, and the directed arcs represent conditional independence assumptions. Given a structure of the network S^h and a set of local conditional probability distributions (CPDs) of each variable X_i, the joint distribution of all the variables can be decomposed as

$$p(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}_s, S^h) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S^h), \qquad (1)$$

where $\mathbf{pa}_i$ denotes the parents of node X_i, and the parameter vector $\boldsymbol{\theta}_s = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$ specifies the CPDs. Our interest in learning task constraints through a BN includes discovering from a complete dataset $\mathbf{D} = \{\mathbf{x}^{1:N}\}$

1. how one variable X_i depends on others $X_{j \neq i}$(the CPDs encoded by $\boldsymbol{\theta}_s$), and
2. what the possibly irrelevant variables $\hat{X}_i$ for a given task are (the conditional independence between variables encoded by S^h).

We note that the former is an instance of parameter learning and the latter of structure learning. Various algorithms and techniques have been developed to learn a BN in different model and data conditions (see [23], [24] for a review). Given that our $\mathbf{X}$ includes both discrete and continuous variables, but the discrete variables do not have continuous parents, we choose to use a *conditional Gaussian network* where exact local computation methods are available [25], [26]. Since the distribution of the continuous variables are not necessarily unimodal Gaussian, we model them as Gaussian mixture nodes.

In this paper, we will not approach the problem of structure learning, but keep it as a topic of future work. Our coarse structure will be suited to the feature subsets presented in Section III, and according to the structure given in Fig. 2.

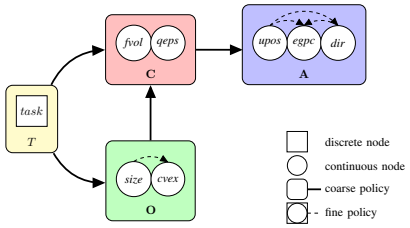


Fig. 2. We follow a coarse policy to generally structure a BN based on feature subset dependencies (in $T, \mathbf{O}, \mathbf{A}, \mathbf{C}$). Our experimental instantiation will in addition use a finer policy, based on specifically extracted features (Section V-C), taking into account dependencies inside the feature subsets.

V. SYSTEM ARCHITECTURE

In this section, we will instantiate an exemplary set $\mathbf{X}$ for task-oriented grasp learning, by selecting and extracting

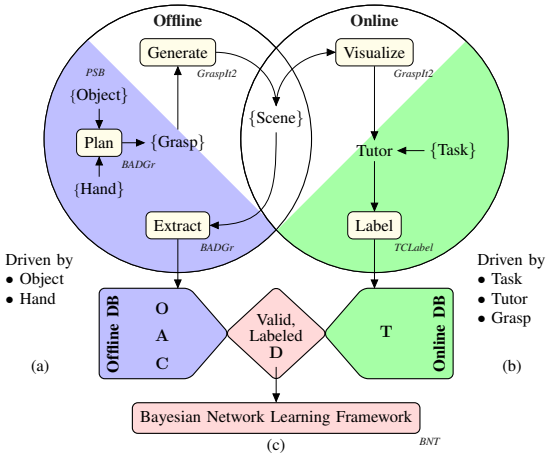


Fig. 3. Complete system architecture including: (a) grasp generation, feature extraction and offline database; (b) visualization, labeling, and online database; (c) Bayesian learning and inference system.

a set of specific features. We first present our architecture that allows object, action and constraint feature extraction from a simulation environment, and labeling of task features through one or more human tutors; then, we will describe in detail the features that we will use for our experiment.

A. Offline Grasp Planning and Feature Extraction

Our complete system architecture is shown in Fig. 3. We use GraspIt! 2.0 [14] as a simulation environment to provide the basis for data generation and visualization of our experiments. We decided on two hand models similar to those in Fig. 1 (a 20-DoF human hand model and a 7-DoF Schunk Dexterous hand model) for generating grasp hypotheses over 25 object models with 6 types taken from the Princeton Shape Benchmark (PSB) [27]. To generate a set of grasp hypotheses on each object-hand pair, we use a planner for *Box Approximation, Decomposition and Grasping* (BADGr) [13]. BADGr can not only be used to extract action features $\mathbf{A}$ from the scene, but also offers several shape representations to build an object feature set $\mathbf{O}$. Since $\mathbf{O}, \mathbf{A}$ are available, we extended the system to generate specified constraint features $\mathbf{C}$. We store, in an offline dataset, $(\mathbf{o}, \mathbf{a}, \mathbf{c})$ as instantiated feature vectors for only those grasps which result in valid force-closure grasps (GraspIt! provides stability criteria to identify those). As a result, this dataset consists of object-hand-driven stable grasp hypotheses, describing object, action and constraint features.

B. Online Task Labeling and Complete Datasets

The scene that corresponds to a grasp hypothesis is visualized in an online framework, as sketched on the right side of Fig. 3. Obviously, it is most efficient to use GraspIt! for that purpose, since it easily allows to reconstruct a grasp configuration from $(\mathbf{o}, \mathbf{a}, \mathbf{c})$, and also provides a tool to interactively explore the scene in 3D. The scene and a set of possible tasks $\mathcal{T}$ is presented to the tutor, who will then label the visualized grasp configuration to be valid or invalid for these tasks. If $(\mathbf{o}, \mathbf{a}, \mathbf{c})$ is labeled to be valid for a task t, a dataset $D_i = (t, \mathbf{o}, \mathbf{a}, \mathbf{c})$ will be included in the training data. Note that, for this reason, our training is based on positive examples, not considering negative (i.e. non-force-closure, or non-labeled) examples.

As a result, the online database consists of grasp identifiers which are labeled with the tasks they satisfy according to human tutoring. The online and offline database are combined to generate a set of training data $\mathbf{D} = \{T, \mathbf{O}, \mathbf{A}, \mathbf{C}\}$ for learning the Bayesian network (see Section IV).

C. Network Instantiation

After presenting the architecture and a coarse network structure policy in Fig. 2, we need to define an experimental set of features $\mathbf{X} = \{T, \mathbf{O}, \mathbf{A}, \mathbf{C}\}$ to evaluate the capabilities of our learning framework. We emphasize that this instantiation is the step that represents the independency of the learning framework per se from different grasp planning systems. Several grasp planners can provide very different representations of grasps and objects. We use the BADGr since it is able to generate a *constrained* number of *intuitive* grasp hypotheses for an object, which makes human labeling effort feasible. Following instances of object and action features are thus adopted from the BADGr planner, while task and constraint features have been selected to demonstrate our task-constraint learning approach. We provide a detailed description of the technical extraction of features in this section, as also an overview in Tab. I. Also, the finer policy, relating some intra-subset features to each other in our BN framework, was already included in Fig. 2.

Task (T): For the current study, we use a single discrete node to represent the task variable with three states $\mathcal{T} = \{\textit{hand-over}, \textit{pouring}, \textit{tool-use}\}$. This decision connects to our choice of object classes from the PSB, summarized in Tab. II as a set of hand-over-able, pour-able, or tool-use-able objects.

Object Size (O_1): A first step of BADGr is to envelop the object's point cloud by a minimum volume bounding box, the so-called 'root' box B_0. The size of the object is taken to be the three dimensions of this box, thus corresponding to width, depth and height of the object. We note that all objects have been adjusted in such a way that their 'top' points in positive z-axis direction (see Tab. II).

Object Convexity (O_2): A subsequent step of BADGr is to decompose the root box B_0 and re-approximate until a fitness measure is reached (details in [13]). After this process, a number n of bounding boxes $B_{0<i<n}$ has emerged which envelop parts of the object. The object's convexity is approximated to be the ratio of volumes before and after the decomposition, as $cvex = volume(B_0) / \sum_{i=1}^{n} volume(B_i)$.

Eigengrasp Pre-Configuration (A_1): Though the pose of each hand model (20-DoF Human hand; 7-DoF Schunk hand) is planned in BADGr, the hand configuration, or pre-shape, is not. As each configuration is embedded in a high-dimensional spaces, we use the idea of [22] to project those into 2D Eigengrasp spaces, where the two dimensions roughly depict spread and extension of each hand. These mappings come with GraspIt: for the human hand, they come with the Columbia Grasp Database (CGDB) [28], which is based on [22]; for the Schunk hand, the first dimension is mapped to the spread joint, and the extension to the three

TABLE I

FEATURE SETS USED IN OUR EXPERIMENTS.

Set	Symbol		Description	Type
	T	*task*	$\mathcal{T} = \{$*hand-over*, *pouring*, *tool-use*$\}$	$\mathbb{D}^3$
O	O_1	*size*	Object (bounding box) size.	$\mathbb{R}^3$
	O_2	*cvex*	Object convexity.	$\mathbb{R}^1$
A	A_1	*egpc*	Eigengrasp pre-configuration.	$\mathbb{R}^2$
	A_2	*upos*	Unified position.	$\mathbb{R}^3$
	A_3	*dir*	Quaternion orientation.	$\mathbb{R}^4$
C	C_1	*fvol*	Free volume.	$\mathbb{R}^1$
	C_2	*qeps*	Grasp Stability.	$\mathbb{R}^1$

TABLE II

SELECTION OF PSB OBJECTS AND CLASSES IN OUR EXPERIMENTS.

Object Type[1]		Selected PSB object IDs	#	z^+-axis
1	*bottle*	483, 484, 490, 493	4	lock
2	*glass*	494, 496, 498	3	top
3	*mug*	504, 507, 508, 509 ($\times$2 scales)	8	top
4	*knife*	718, 720, 724	3	point
5	*hammer*	1109, 1110, 1111, 1112	4	head
6	*screwdriver*	1113, 1114, 1115	3	head

proximal finger joints. For each grasp pose that BADGr generates, we sample 5 random Eigengrasps.

Unified Grasp Position (A_2): The grasp approach direction towards the object will be an important feature of our experiment. For example, one could imagine a constraint that a mug should not be grasped from above in case of a pouring task. We note three aspects to account for: (i) it depends greatly on the object shape where BADGr will generate grasps, thus the 3D space will be covered only very sparsely, (ii) not only the position, but also the approach orientation will affect where an object will be grasped, and (iii) it should not matter from how far a grasp is triggered. To approach all these issues, we project the 3D grasp position to a 2D spherical space. The projection sphere is defined by the center point of the specific object (taken from the root box) and a fixed radius (which only has to ensure that all objects are inside this radius). We then intersect this sphere with the grasp approach vector emitted from the grasp position. We call the intersection point the *unified grasp position*.

Grasp Orientation (A_3): Though the unified grasp position is using the grasp direction, it is not encoding it. Thus, we consider this value in a separate variable. The grasp orientation is embedded in each grasp generated by BADGr, in terms of a quaternion representation.

Free Volume (C_1): The free-volume constraint defines the percentage of the object volume that corresponds to the non-covered part in a grasp configuration. Briefly, we span a tetrahedron Δ using the palm position and the three contact points that maximize the volume of Δ. Considering the box decomposition $B_{0<i<n}$, we compute to what extent Δ intersects each B_i, and sum up the volumes to V_{occ} (occupied volume). The free volume is then acquired as $fvol = 1 - V_{occ} / \sum_{i=1}^{n} volume(B_i)$.

Grasp Stability (C_2): To incorporate force-related task constraints, we use one of the commonly used measures of grasp stability that GraspIt! provides, eps_{L1}. It describes stability of each grasp in terms of force-closure (see [29]).

D. Bayesian Network Learning Framework

To train and use the BNs (Fig 2) for human and Schunk hands, we use the BNT [30], the Bayes Net Toolbox for Matlab. In the following experiments, the training data comes from the on-line labeling by only one human expert. For the human-hand BN, the training set includes 600×3 instances, with 600 instances per task, and around 100 instances per object type; and for the Schunk-hand BN, the training set includes 1200×3 instances, with 1200 instances per task, and around 200 instances per object type. The testing set comes from the 6 objects that are not included in the training set. Each of the 6 objects belongs to one of the 6 object types. This is to evaluate how well the trained network can generalize to the unknown objects.

At this point, we would like to refer the reader to the accompanying video providing a practically focussed visualization of the described grasp generation and architecture.

VI. EXPERIMENTAL RESULTS

In this section, we will describe the application of the trained BN for three different experiments. While two of them will mainly provide a view on the evaluation of the technique, the third one will show a setup for robot imitation based on task-constraints. For each experiment, we formulate the corresponding semantic questions to the system.

A. *"From where to grasp an object, given a task?"*

Formulating this question as $P(upos|task,\ size,\ conv)$, our goal is to observe how our three tasks influence the *position* of a grasp, *upos*. Note that *upos* only provides the information on 'where' the hand can be placed with respect to the object to fulfill a task. It does not encode the complete information on 'how' to grasp the object, which needs a combination of all the action variables to represent (see Section V-C).

As representatives for the experimental results, we select a hammer, a bottle, and a mug out of the 25 object models (see Tab. II) as the test set, and train the Bayesian network using the Schunk hand data stored from the remaining 22 models. We then compute P for all 3 test objects, and all 3 tasks. The results for this experiment are shown in Tab. III.

Analyzing the results, we can make the following observations: (i) the BN is clearly affected by the BADGr planner, providing a lot of "from where to grasp" hypotheses from the four sides, top and bottom of an object. (ii) Given a *hand-over* task, the results do not substiantially differ, and all major directions are valid. (iii) Given a *pouring* task, the network clearly rejects to grasp from the top in cases of bottle and mug. That also the hammer has some (but much less) likelihood to be poured from these directions, is grounded in our only object features of size and convexity; the hammer

[1]In an electronic version, entries in this column are linked to the specific PSB [27] object categories at http://shape.cs.princeton.edu/benchmark/.

TABLE III

EXPERIMENT IV.A: DISTRIBUTION OF UNIFIED POSITION CONDITIONED ON TASKS AND OBJECT FEATURES $P(upos|T, size, conv)$.

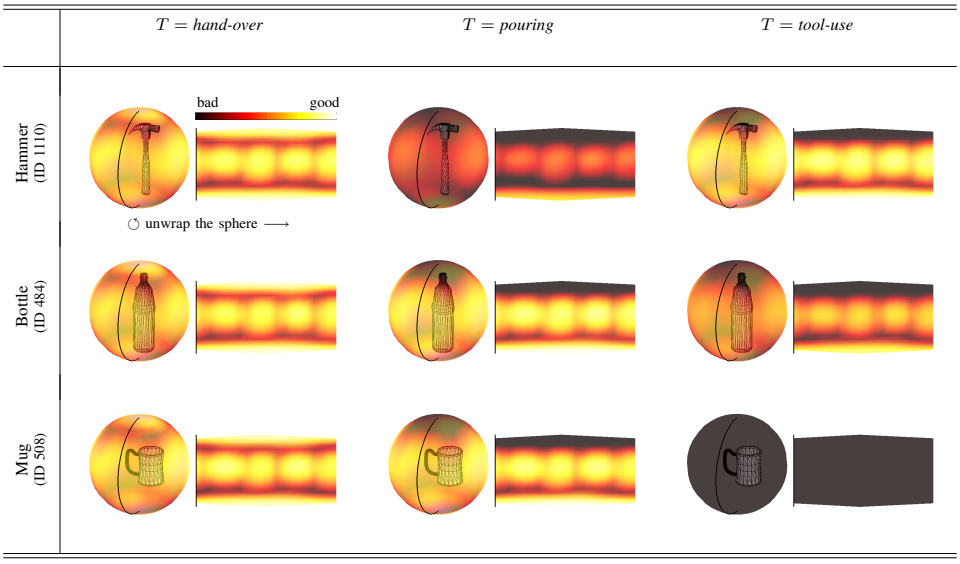
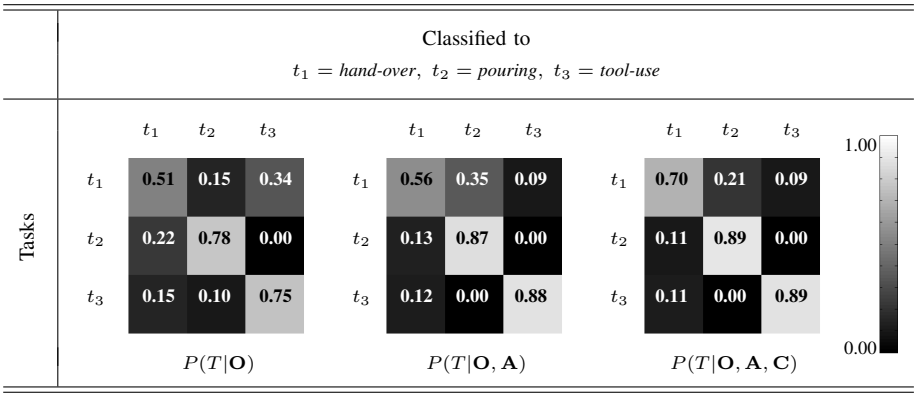


TABLE IV

EXPERIMENT IV.B: CROSS-VALIDATION OF TASK CLASSIFICATION GIVEN OBJECT, ACTION AND / OR CONSTRAINT FEATURES.



has similar size as a bottle, but higher convexity. (iv) For *tool-use*, the network emphasizes the hammer, from sides and bottom, to be tool-use-able. It correctly rejects grasps from the top. In a same way, and for the same reasons as in the "pourable hammer" case, the bottle is tool-usable. The mug is identified as being non-tool-use-able at all, since it is very much different in size and convexity from a usual tool (which should be long and convex up to some extent).

B. *"What tasks is this (object / object and action / object, action and constraint) good for?"*

Dependent on the characteristic of *this*, the question can be formulated as $P(T|\mathbf{O})$, $P(T|\mathbf{O}, \mathbf{A})$, or $P(T|\mathbf{O}, \mathbf{A}, \mathbf{C})$. Since the task is represented by a single discrete node, we can identify each problem as a classification, given different amounts of observations. In this experiment, our goal is to analyze how good these classifications work for unknown objects. We train the networks for the human and the Schunk hand, leaving out 1 object per object type. Thus, in both cases, our training set includes data from 19 objects with the test data covering all 6 object types. We compute an average classification rate over all three P. The results for this experiment are shown in Tab. IV.

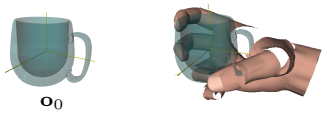
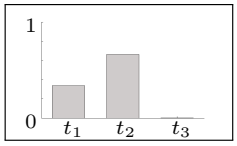
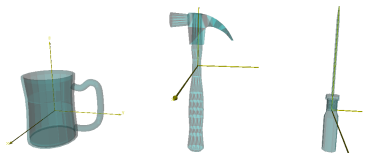
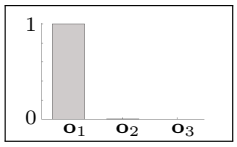
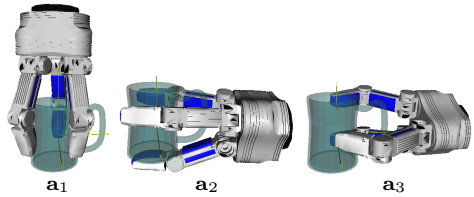
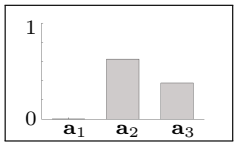
Analyzing the results, we can make the following observations: (i) object features contain important information for task, in particular *pouring* (78% classification rate) and *tool-use* (75%); (ii) introduction of action features improves correct classification of these tasks (87% and 88%), but does not affect *hand-over* (56%) significantly; (iii) when introducing constraint features, *pouring* and *tool-use* do not improve significantly (89% and 89%), but *hand-over* (70%). These numbers correspond to correct task classification rates, given an unknown model, an action, and the encoded constraints.

C. *"Can you imitate this grasp?"*

In this section, we demonstrate the use of the task constraint Bayesian network in a goal-directed imitation experiment (see Tab. V). The experiment is implemented using the human hand model as the demonstrator, and the Schunk hand as the imitator. We therefore train the networks for both hands, letting out the four test objects $\mathbf{o}_0$ to $\mathbf{o}_3$ presented in Tab. V. The goal is to imitate the demonstrator

TABLE V

EXPERIMENT IV.C: GOAL-DIRECTED IMITATION ON *'pouring'* TASK.

	Scenes	Probabilities	Decisions
Step 1 (human demo)	 $\mathbf{o}_0$	$P^H(T\|\mathbf{O},\mathbf{A},\mathbf{C})$ 	$\hat{t}^H = t_2$ (*pouring*)
Step 2.1 (select object among $\mathcal{O}$)	 $\mathbf{o}_1$ $\mathbf{o}_2$ $\mathbf{o}_3$	$P^R(T=t_2\|\mathbf{O})$ 	$\mathbf{o}^* = \mathbf{o}_1$ (*mug*)
Step 2.2 (select action among $\mathcal{A}$)	 $\mathbf{a}_1$ $\mathbf{a}_2$ $\mathbf{a}_3$	$P^R(T=t_2\|\mathbf{o}_1,\mathbf{A},\mathbf{C})$ 	$\mathbf{a}^* = \mathbf{a}_2$ (*stable side grasp*)

performing a *pouring* task using a mug $\mathbf{o}_0$. We first describe the general formulation in a two-step imitation framework:

In the first step, the robot observes a human performing a grasp on an object, and estimates the intention (task) t^H of the human action. The probability of the tasks for the demonstrated object-grasp combination is encoded by the task-constraint BN specific to the demonstrator's embodiment $P^H(T|\mathbf{O},\mathbf{A},\mathbf{C})$. We denote the maximum-likelihood estimate of the task as $\hat{t}^H$.

In the second step, the robot finds the most compatible grasp on the object (or objects) it perceived, in order to achieve the same task $\hat{t}^H$. This step can be formulated as a Bayesian decision problem, where a reward function r defines the degree of similarity in the set of features $\mathbf{x} = \{\mathbf{o},\mathbf{a},\mathbf{c},t\}$ between the demonstrator and the robot. Here $\mathbf{o},\mathbf{a},\mathbf{c},t$ represent the instantiated object, action, constraint and task features corresponding to the network variables $\mathbf{O},\mathbf{A},\mathbf{C},T$. As the knowledge over this feature set is not certain, the expectation $E()$ is taken over the reward function. For instance, the probability of the suitable task given an object-grasp combination is encoded by the task constraint BN specific to the robot's embodiment $P^R(T|\mathbf{O},\mathbf{A},\mathbf{C})$. The general optimization function for decision making is then

$$\langle \mathbf{a}^*, \mathbf{o}^* \rangle = \underset{\mathbf{a}\in\mathcal{A},\mathbf{o}\in\mathcal{O}}{\operatorname{argmax}} E\Big(r(\mathbf{a}^H,\mathbf{o}^H,\mathbf{c}^H,\hat{t}^H,\mathbf{a},\mathbf{o},\mathbf{c},t)\Big)\ , \quad (2)$$

where superscript ^{H} indicates features from human demonstration. The maximization is over a set of stable grasp hypotheses $\mathcal{A} = \{\mathbf{a}_1,\ldots,\mathbf{a}_{n_a}\}$ generated by the robot's grasp planner, and / or available objects $\mathcal{O} = \{\mathbf{o}_1,\ldots,\mathbf{o}_{n_o}\}$ presented to the robot. We present two experiments to illustrate the fomulation:

1) Task Goal Matching: The objective is to plan a grasp to match the same task goal while the robot is given a *single object*, a mug $\mathbf{o}_1$, suitable for the task (see Tab. V). In step 1, the robot estimates the most likely task of the demonstrated grasp to be *pouring* $\hat{t}^H = t_2$. In step 2.2, the reward is a simple indicator function of the demonstrated task, $r(t_2)$, which equals 1 if $t = t_2$, 0 otherwise. The optimization function is simply:

$$\langle \mathbf{a}^* \rangle = \underset{\mathbf{a}\in\{\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3\}}{\operatorname{argmax}} rP^R(t=t_2|\mathbf{o}=\mathbf{o}_1,\mathbf{a},\mathbf{c})\ . \quad (3)$$

As a result, the robot selects $\mathbf{a}_2$, a side grasp that looks more stable for the pouring task. It is worth noticing that, with the BN as the knowledge base, the robot can not only plan grasps according to the 'more obvious' *not-cover-opening* constraint through rejecting the first hypothesis $\mathbf{a}_1$. It can also prefer the side grasp with a specific object-hand configuration that could better afford the subsequent pouring action.

2) Object Selection and Task Goal Matching: In this more complex scenario, the robot is confronted with multiple objects: a mug, a hammer and a screw-driver, represented by an object set $\mathcal{O} = \{\mathbf{o}_1,\mathbf{o}_2,\mathbf{o}_3\}$. In the second imitation step, the robot first follows step 2.1 to select the mug $\mathbf{o}_1$, and then step 2.2 to select the best grasp action. The reward function is the same as in the task goal matching example, but the optimization is also computed over the available objects,

$$\langle \mathbf{a}^*, \mathbf{o}^* \rangle = \underset{\mathbf{a}\in\mathcal{A},\mathbf{o}\in\mathcal{O}}{\operatorname{argmax}} rP^R(t=t_2|\mathbf{a},\mathbf{o},\mathbf{c})\ . \quad (4)$$

As is shown in Tab. V, the network clearly rejects the hammer and the screw-driver, since the features $\mathbf{o} = \{size, cvex\}$ of the objects clearly separate the tasks they afford between *pouring* and *tool-use*.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a probabilistic framework for learning of task constraints in grasp selection. Our constraint learning model links the semantic requirements of manipulation tasks to the continuous feature space of the objects and grasp actions. Our approach is semi-automated and embodiment-specific. A simulation-based grasp planner generates a set of hand-specific, stable grasp hypotheses on a range of objects. A teacher provides the knowledge of task requirements by labeling each hypothesis with the suitable manipulation task(s). The underlying relations between the conceptual task goals and the continuous object-action features are encoded by the probabilistic dependencies in a Bayesian network. Using this network as a knowledge base, the simulation experiments showed that the robot is able to infer the intended task of a human demonstration, choose the object that affords this task, and select the best grasp action to fulfill the task requirements. Though we implement and test the current framework based on the BADGr grasp planner [13], this task constraint framework can be integrated with any grasp planning system.

In the current implementation, we do not address the learning of the network structure, but manually connect the nodes based on expert knowledge. In the future, we intend to introduce more tasks, constraint functions, as well as different and potentially redundant object and action features. In such cases, building the network structure based purely on human knowledge will be cumbersome and nonreliable. Data-driven, automated structure learning is needed to identify the task-relevant variables, and discover the underlying dependencies between these variables.

In addition, we would like to implement the approach in the real robot sensory-motor platforms. For example, we can introduce different object representations by applying different vision modules, allowing the network to encode uncertainty in the perception system. Further, more complex conditional probability densities allow modeling of more complex relationships between the variables. We also plan to introduce a dynamic Bayesian network to incorporate robot self-exploration with human-based learning to incrementally enrich the grasp-related knowledge of the world.

ACKNOWLEDGMENTS

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

REFERENCES

[1] D. Berenson, S. Srinivasa, and J. Kuffner, "Addressing Pose Uncertainty in Manipulation Planning Using Task Space Regions," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1419–1425.

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

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

[4] D. B. Grimes and R. P. N. Rao, "Learning Actions through Imitation and Exploration: Towards Humanoid Robots that Learn from Humans," in *Creating Brain-Like Intelligence*, ser. Lecture Notes in Computer Science, vol. 5436. Springer, 2009, pp. 103–138.

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

[6] C. Acosta-Calderon and H. Hu, "Robot Imitation: Body Schema and Body Percept," *Applied Bionics and Biomechanics*, vol. 2, no. 3-4, pp. 131–148, 2005.

[7] D. Jain, L. Mösenlechner, and M. Beetz, "Equipping Robot Control Programs with First-order Probabilistic Reasoning Capabilities," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3130–3135.

[8] M. Toussaint, N. Plath, T. Lang, and N. Jetchev, "Integrated Motor Control, Planning, Grasping and High-level Reasoning in a Blocks World using Probabilistic Inference," in *IEEE International Conference on Robotics and Automation*, 2010, to appear.

[9] C. L. Nehaniv and K. Dautenhahn, Eds., *Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press, 2004.

[10] D. Wolpert and M. Kawato, "Multiple Paired Forward and Inverse Models for Motor Control," *Neural Networks*, vol. 11, no. 7-8, pp. 1317–1329, October 1998.

[11] Y. Demiris and M. Johnson, "Distributed, Predictive Perception of Actions: A Biologically Inspired Robotics Architecture for Imitation and Learning," *Connection Science*, vol. 15, no. 4, pp. 231–243, 2003.

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

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

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

[15] Z. Xue, A. Kasper, M. J. Zoellner, and R. Dillmann, "An Automatic Grasp Planning System for Service Robots," in *14th International Conference on Advanced Robotics*, 2009.

[16] T. Baier and J. Zhang, "Reusability-based Semantics for Grasp Evaluation in Context of Service Robotics," in *IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 703–708.

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

[18] A. Rao, B. A. Olshausen, and M. Lewicki, Eds., *Probabilistic Models of the Brain: Perception and Neural Function*. MA: MIT Press, 2002.

[19] R. Zöllner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards Cognitive Robots: Building Hierarchical Task Representations of Manipulations from Human Demonstration," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1535–1540.

[20] M. Novotni and R. Klein, "Shape Retrieval using 3D Zernike Descriptors," *Computer-Aided Design*, vol. 36 (11), pp. 1047–1062, 2004.

[21] T. Feix, R. Pawlik, H.-B. Schmiedmayer, J. Romero, , and D. Kragic, "A Comprehensive Grasp Taxonomy," Poster Presentation at Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, June 2009.

[22] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous Grasping via Eigengrasps: A Low-dimensional Approach to a High-complexity Problem," in *RSS 2007 Manipulation Workshop*, 2007.

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

[24] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, M. Jordan, S. L. Lauritzen, J. F. Lawless, and V. Nair, Eds. Springer-Verlag New York, 1999.

[25] S. L. Lauritzen and F. Jensen, "Stable Local Computation with Conditional Gaussian Distributions," *Statistics and Computing*, vol. 11, no. 2, pp. 191–203, 2001.

[26] S. G. Bøttcher, "Learning Bayesian Networks with Mixed Variables," Ph.D. dissertation, Aalborg University, DK, 2004.

[27] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *International Conference on Shape Modeling and Applications*, 2004, pp. 167–178.

[28] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The Columbia Grasp Database," in *International Conference on Robotics and Automation*, 2009, pp. 3343–3349.

[29] C. Ferrari and J. Canny, "Planning Optimal Grasps," in *IEEE Int. Conference on Robotics and Automation*, vol. 3, 1992, pp. 2290–2295.

[30] K. Murphy, "BNT – Bayes Net Toolbox for Matlab," **[URL]** http://code.google.com/p/bnt/, 1997. Last visited March 14, 2010.

9th IEEE-RAS International Conference on Humanoid Robots
December 7-10, 2009 Paris, France

Rapid Learning of Humanoid Body Schemas with Kinematic Bézier Maps

Stefan Ulbrich*, Vicente Ruiz de Angulo†, Tamim Asfour*, Carme Torras† and Rüdiger Dillmann*
* Institute for Anthropomatics,
Karlsruhe Institute of Technology, Germany
Email: [ulbrich,asfour,dillmann]@ira.uka.de
† Institut de Robòtica i Informàtica Industrial,
CSIC-UPC, Barcelona
Email: [ruiz,torras]@iri.upc.edu

Abstract— **This paper addresses the problem of hand-eye coordination and, more specifically, tool-eye recalibration of humanoid robots. Inspired by results from neuroscience, a novel method to learn the forward kinematics model as part of the body schema of humanoid robots is presented. By making extensive use of techniques borrowed from the field of computer-aided geometry, the proposed *Kinematic Bézier Maps (KB-Maps)* permit reducing this complex problem to a linearly-solvable, although high-dimensional, one. Therefore, in the absence of noise, an *exact* kinematic model is obtained. This leads to rapid learning which, unlike in other approaches, is combined with good extrapolation capabilities. These promising theoretical advantages have been validated through simulation, and the applicability of the method to real hardware has been demonstrated through experiments on the humanoid robot ARMAR-IIIa.**

I. INTRODUCTION

With increasingly complex robots –especially humanoids– the calibration process of the arms and other kinematic chains, and hence the prediction of the effects of joint movements, becomes a difficult, time-consuming and often expensive task. This process has to be repeated every time the tool center point (TCP) of the robot changes, e.g. if the robot accidently suffers deformation or –even more important– if the robot intends to interact with its environment with a tool. The hand-eye calibration by traditional means then becomes nearly impossible. Humans solve the problem successfully by pure self-observation, which has led to the adaptation of biologically-inspired mechanisms to the field of robotics.

In neuroscience, it is common knowledge that there exists a body schema that correlates proprioceptive sensor information, e.g. joint configurations, with the visible shape of the body [10]. It also represents an unconscious awareness of the current body state [11]. Experiments with both macaque monkeys and humans showed that the body schema is neither congenital nor rigid but rather learnable and adaptable, as shown by Maraviata et al [9]. For instance, an experiment examines the proximal visual receptive field (the area in cartesian space where stimuli activate the neurons associated to grasping) of the macaque monkeys. It was shown that this field was enlarged by the length of a tool that the monkeys used once they had been trained to do so. This leads to the conclusion that the tool itself became incorporated into the monkey's own body schema. Similar conclusions were drawn from experiments with human patients who suffered from brain damage or phantom pain after having lost a limb. This leads to the assumption that in the human brain similar processes exist as in the monkey's. Further observations by Stamenov [14] showed that the body schema is not a well-formed pattern but rather a set of several connected groups of neurons that represent opportunistically learned manifolds and that are distributed over regions in the brain.

As a consequence of these results, there is a great interest among robotics researchers to emulate this adaptability with techniques from machine learning. In most robotics works, the term 'learning of the body schema' is restricted to the sub-symbolic learning of the relation between the proprioceptive sensors for the joint configuration $\boldsymbol{\theta}$ and the visual position $\boldsymbol{x}$ of the end-effector. Therefore, it is basically limited to the approximation of the forward kinematics (FK), the inverse and local inverse kinematics (IK) from pairs of joint angles and cartesian coordinates:

$$f(\boldsymbol{\theta}) = \boldsymbol{x}, \quad f^{-1}(\boldsymbol{x}) = \boldsymbol{\theta} \text{ and } f^{-1}(\dot{\boldsymbol{x}}) = \dot{\boldsymbol{\theta}}. \qquad (1)$$

In general, the approximation of the latter two functions (with a high number of DoF) is an ill-posed problem as the same position can be generated by different joint configurations. However, the approximation of the FK can be used to solve the IK problem in a flexible way via techniques such as *resolved motion rate control (RMRC)* [16]. Thus, the current paper focuses on learning the FK mapping from tuples $(\boldsymbol{\theta},\boldsymbol{x})$, which will be referred to as *training experiences, samples* or *training data.*

The main difficulty of the approximation of the FK lies in the fact that it is a highly non-linear function with non-redundant input variables, each of them significantly

The work described in this paper was partially conducted within the EU Cognitive Systems projects GRASP (FP7- 215821) and PACO-PLUS (FP6-027657) funded by the European Commission.
The authors acknowledge support from the Generalitat de Catalunya under the consolidated Robotics group, and from the Spanish Ministry of Science and Education, under the project DPI2007-60858.

influencing the result. Hence, it requires a large amount of training experiences that grows exponentially with the number of DoF of the kinematic chain. This complexity can be reduced by decomposing the robot into kinematic subchains as proposed by Ruiz et al. [2][3], but at the expense of increasing the demands on the robot's perceptive abilities or limiting the applicability to a family of robots.

Parametrized Self-Organizing Maps (PSOMs) [15] have been often used to learn kinematics problems because of its versatility and interpolation abilities. However, they require that the training samples are distributed in a regular grid (although this is mitigated in [8]) and, specially, they are not well suited to on-line learning. High-dimensional kinematic chains have been handled by using *locally weighted projection regression (LWPR)* [4]. This algorithm creates linear models locally valid for the training data, which are combined into a weighted sum that eventually approximates the FK or local IK. PSOM has in common with the LWPR approach that they quickly produce locally valid approximations but again require a large amount of training data for a complete model, as they lack good extrapolation capabilities. An exact encoding of the FK of robots with rotational joints is not possible as both approaches use approximations that are not capable of describing the product space of rotations with a finite number of samples. However, both can be used to learn a local IK approximation and are thus capable of solving the IK problem directly.

A different approach was recently proposed by Hersch et al [7], where the parameters of the FK in Denavit-Hartenberg convention are learned directly by an optimization algorithm. This optimization eventually leads to the creation of a body schema with good extrapolation capabilities and even converges to an exact model in simulation. However, this method suffers from a low learning speed –even in simulation.

To the best of our knowledge, there is not yet an algorithm that can learn a FK mapping *exactly* and in an efficient way. This is the aim of this work, where we use techniques from the field of Computational Geometry –namely, rational Bézier tensor-product functions. Derived from these functions the *Kinematic Bézier Maps (KB-Maps)* were created. In contrast to all other approaches, this representation permits an exact encoding of the FK, which is robust to sensor noise, and it allows the learning algorithm to keep the same complexity regardless of the number of training experiences. Moreover, it exhibits good extrapolation capabilities even when only a relatively small number of experiences can be provided that lie close to one another. The key aspect of the KB-Maps is that they transform a highly non-linear problem into a higher-dimensional, but linearly solvable, equation system.

The paper is structured as follows. In the next section, a brief introduction to the underlying geometrical techniques is provided. Section 3 describes their application in the KB-Maps to encode FK. Two algorithms suitable to perform the learning are presented in Section 4. Then, in Section 5, the proposed method is applied to the humanoid robot ARMAR-IIIa [1] in both real experiments and simulation, and the obtained results are discussed. The paper concludes with a brief account of the contributions and an outlook on future work.

II. Forward Kinematics Representation in Bézier Form

A. Mathematical Fundamentals

1) Bézier Curves: In affine space, every polynomial spatial curve $\boldsymbol{b}(s)$ of degree n has an unique Bézier form [13] [6]:

$$\boldsymbol{b}(s) = \sum_{i=0}^{n} \boldsymbol{b}_i \cdot B_i^n(s), \text{ with } B_i^n(s) := \binom{n}{i} \cdot s^i \cdot (1-s)^{n-i}, \quad (2)$$

where every point $\boldsymbol{b}(s)$ on the curve is the result of an affine combination of a set of $n+1$ *control points* $\boldsymbol{b}_i$ weighted by the well-known *Bernstein polynomials* $B_i^n(s)$ that serve as a basis for all polynomial curves of degree n. The Bézier form of the curve's derivative

$$\dot{\boldsymbol{b}}(s) = n \cdot \sum_{i=0}^{n-1} \Delta \boldsymbol{b}_i \cdot B_i^{n-1}(s) \quad (3)$$

can be obtained easily by the construction of the forward differences $\Delta \boldsymbol{b}_i$ with

$$\Delta \boldsymbol{b}_i := \boldsymbol{b}_{i+1} - \boldsymbol{b}_i.$$

2) Tensor Product Bézier Surfaces: Polynomial surfaces and higher multivariate functions can also be expressed in Bézier form. If they are polynomial of degree n in their main directions (when only one parameter is variable), the function can be expressed as a tensor product of two or more Bézier curves. For example, a polynomial surface of degree n, $\boldsymbol{b}(s_1, s_2)$, has the tensor product Bézier form

$$\boldsymbol{b}(s_1, s_2) = \sum_{i_1=0}^{n} \cdot \left(\sum_{i_2=0}^{n} \boldsymbol{b}_{i_1,i_2} \cdot B_{i_2}^n(s_2) \right) \cdot B_{i_1}^n(s_1). \quad (4)$$

The net of $(n+1)^2$ points $\boldsymbol{b}_{i_1,i_2}$ forms the *control net.* In general, a d-dimensional tensor product Bézier of degree n can be represented as

$$\boldsymbol{b}(\boldsymbol{s}) = \sum_{\boldsymbol{i}} \boldsymbol{b}_{\boldsymbol{i}} \cdot B_{\boldsymbol{i}}^n(\boldsymbol{s}) \quad (5)$$

where $\boldsymbol{i} := (i_1, i_2, \ldots, i_d)$ represents a vector of indices going through the set $\mathscr{I}_n = \{(i_1, i_2, \ldots, i_d)\ s.t.\ i_k \in \{0, \ldots, n\}\}$ of index vectors addressing the points of the control net , $\boldsymbol{s} := (s_1, s_2, \ldots, s_d)$ is the parameter vector, and

$$B_{\boldsymbol{i}}(\boldsymbol{s}) := \prod_{k=1}^{d} B_{i_k}(s_k) \quad (6)$$

are the products of all Bernstein polynomials within each summand. In total, the control net of the tensor product Bézier representation is formed by $(n+1)^d$ control points.

3) Rational Polynomials and Rational Bézier Form: Although FK can be approximated by polynomials, an exact representation of the FK requires a more complex class of functions, e.g. *rational polynomials* [5]. Rational polynomial functions are similar to affine polynomial functions except for the fact that they are defined in the *projective space* $\mathscr{P}$. Simplifying, $\mathscr{P}$ is a space with an additional dimension and elements of the form

$$\mathbb{p} = \begin{bmatrix} \gamma \boldsymbol{p} \\ \gamma \end{bmatrix} \text{ or short } \mathbb{p} = \gamma \cdot \boldsymbol{p}, \quad \gamma \in \mathbb{R} \setminus 0,$$

where $\boldsymbol{p}$ is an affine point and γ is called *homogeneous coordinate* or *weight* of $\mathbb{p}$. Any projective point $\mathbb{p} \in \mathscr{P}$ can be understood as a ray that originates from the the *projective center* $(0,\ldots,0)$ and intersects the affine space at $\boldsymbol{p}$ when $\gamma = 1$. The intersection point is called the *affine image* of $\mathbb{p}$ and division by γ is called *projection (onto the affine space).*

On projection into the affine space, rational polynomials generally become more complex functions and may loose their polynomial characteristics (see Fig. 1). Still, in homogeneous space, there does exist the same previously introduced unique Bézier form for curves and surfaces

$$\mathbb{b}(\boldsymbol{s}) = \sum_i \mathbb{b}_i \cdot B_i(\boldsymbol{s}) = \begin{bmatrix} \boldsymbol{b}(\boldsymbol{s}) \\ \gamma(\boldsymbol{s}) \end{bmatrix} = \begin{bmatrix} \sum_i \gamma_i \boldsymbol{b}_i \cdot B_i(\boldsymbol{s}) \\ \sum_i \gamma_i \cdot B_i(\boldsymbol{s}) \end{bmatrix}.$$

and, after affine projection, the rational Bézier form

$$\mathbf{b}(\boldsymbol{s}) = \frac{\boldsymbol{b}(\boldsymbol{s})}{\gamma(\boldsymbol{s})} = \frac{\sum_i \gamma_i \cdot \boldsymbol{b}_i \cdot B_i(\boldsymbol{s})}{\sum_i \gamma_i \cdot B_i(\boldsymbol{s})}. \tag{7}$$

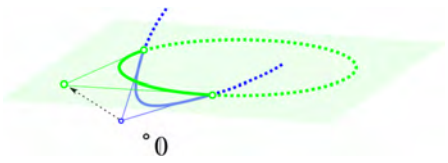


Fig. 1. The projection of a parabola in $\mathscr{P}$ onto a circle.

B. Forward Kinematics Representation: The One-dimensional Case

In this section, we show how to use the techniques presented above to come up with the Bézier representation of the forward kinematics of a robot with rotational joints.

The end-effector of a single-joint ideal robot moves along a circular trajectory when the value θ of its joint changes. In general, the FK of a robot with d degrees of freedom is simply a composition of d circles. Therefore, the basic geometric objects that we need to represent are circles and more generally their deformations. The only deformation of circles that we consider are ellipses. We expect that this flexibility contributes to a better conformation to the real function that has to be learned, that may be biased by the sensorial system or gravity.

To explain more clearly our representation of FK, we begin by showing it for a single degree of freedom. As declared before, our model is able to represent a family of ellipses including the circle.

Homogeneous polynomials of degree two become conics when projected onto the affine space and, for every conic, there exists a rational Bézier representations of degree two [5]. In particular, a rational Bézier curve

$$\mathbf{b}(s) = \frac{\sum_{i=0}^{2} \gamma_i \cdot \boldsymbol{b}_i \cdot B_i^2(s)}{\sum_{i=0}^{2} \gamma_i \cdot B_i^2(s)} \tag{8}$$

is an ellipse if

1) the weights γ_0 and γ_2 are equal, and
2) $\gamma_1/\gamma_0 = \gamma_1/\gamma_2 < 1$.

To be a circle, in addition it has to satisfy that a) the control points form an isosceles triangle with a common angle α, and b) $\gamma_1/\gamma_0 = \cos\alpha$. Note that all conditions refer to proportions between weights because multiplying every weight by a constant leaves (8) unchanged.

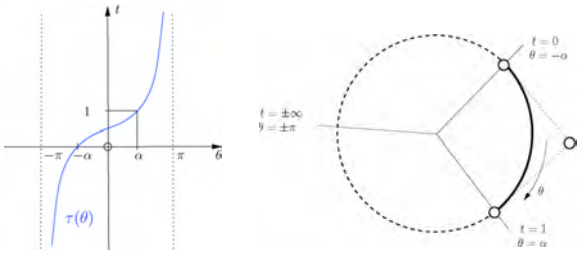
Imposing $\gamma_0 = \gamma_2 = 1$ and fixing γ_1 to an arbitrary constant smaller than one, the ellipse conditions are satisfied. At the same time, doing this, the circle is not excluded from the family of ellipses potentially represented by the Bézier form, since for any γ_1 it is possible to find a set of control points forming an isosceles triangle with a common angle whose cosine is γ_1. Thus, if learning data comes from a circle and we have enough points to constrain the model, we will obtain a circle. By imposing $\gamma_0 = 1$, the redundancy in the representation induced by proportionality in the weights is eliminated. Imposing $\gamma_0 = \gamma_2$ and fixing γ_1 to a constant has the effect of limiting the kind of ellipses that can be used to fit the FK data.

The joint effect of these constraints is that the number of sample points required to determine the Bézier form is greatly reduced (see Section III): in the one-dimensional case, it is reduced from 5 (required in general for an ellipse) to 3. Note that this is also the minimum number of sample points required if we would have assumed a model based only on circles. As a consequence, we have a more flexible model without having to pay a tribute in increased number of required data.

Our model is still incomplete. For $\mathbf{b}(s)$ to represent a complete ellipse, s must go from $-\infty$ to ∞. Instead, the data samples and the robot commands are joint encoder values θ, ranging from $-\pi$ to π. We must transform θ before being used as input to the Bézier form. We have chosen the following transformation

$$\tau : [-\pi, \pi] \mapsto \mathbb{R}, \quad \tau(\theta) = \frac{\tan(\theta/2)}{2 \cdot \tan(\alpha/2)} + \frac{1}{2}. \tag{9}$$

where $\alpha = \arccos(\gamma_1)$, see Fig. 2(a). In fact, it is more practical to fix indirectly γ_1 by choosing first an arbitrary α and setting $\gamma_1 = \cos(\alpha)$. The sense of this transformation is that, when $\mathbf{b}(s)$ becomes exactly a circle, α becomes the common angle in the isosceles triangle formed by the control points, see Fig. 2(b). In this case, it can be proven that θ becomes the angular parameterization of the circle measured in radian units in $\mathbf{b}(\tau(\theta))$, which is the final form of the one-dimensional KB-Maps.



(a) The τ transformation (b) The parameterization

Fig. 2. Transformation from a joint angle to the corresponding parameter of the Bézier form.

C. Forward Kinematics Representation: The Multidimensional Case

We like to represent a composition of d ellipses with a Bézier form, understood in the same sense that a pure FK is a composition of d circles: when all variables but one are fixed the resulting curve must be an ellipse, i.e., the isoparametric curves of the Bézier form are ellipses. To accomplish this, we set the weights $\gamma_{i_1,i_2,\ldots,i_d}$ of control points $\mathbb{b}_{i_1,\ldots,i_d}$ to $\gamma^{ones(i_1,\ldots,i_d)}$, where $ones()$ returns the number of ones in the arguments and γ is an arbitrary constant minor than one. The proof is in the Appendix. The value γ can be selected like in the one-dimensional case, via the cosine of an arbitrary angle, $\gamma = \cos\alpha$.

With arguments similar to those for the one-dimensional case, we can state that each of the ellipses defined by the isoparametric curves in the main directions can take the shape of a circle. Therefore, if we have enough data points to determine the surface (3^d, see Section III) coming from an exact FK, the Bézier form will reproduce exactly the robot kinematics. In that case, the implicit control points (named $\mathbf{q}_k$ in the Appendix) appearing in the expression of the isoparametric curves in the main directions will form an isosceles triangle. In fact, the triangles will be congruent for all main directions, having all the same common angle α. But, of course, the circles in the main directions are anyway unrelated and can be completely different.

Finally, to complete the model we must include the transformation $\tau(\boldsymbol{\theta})$ of the input encoder vector, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$. The rationale is, as in the one-dimensional case, to establish a correspondence between the encoder values that are given in uniform angular units (radians) and the Bézier parameters $\boldsymbol{s}$ that yield the adequate Bézier surface points in the context of an exact FK. In sum, this is the KB-Maps model for FK:

$$\boldsymbol{f}(\boldsymbol{\theta};\boldsymbol{G}) \equiv \mathbb{b}(\tau(\boldsymbol{\theta})) = \frac{\sum_{\boldsymbol{i}} \gamma_{\boldsymbol{i}} \cdot \boldsymbol{b}_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}))}{\sum_{\boldsymbol{i}} \gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}))} \qquad (10)$$
$$\gamma_{\boldsymbol{i}} = \gamma^{ones(\boldsymbol{i})}, \gamma < 1$$

which is the projection onto the affine space of

$$\mathbb{f}(\boldsymbol{\theta};\boldsymbol{G}) \equiv \mathbb{b}(\tau(\boldsymbol{\theta})) = \sum_{\boldsymbol{i}} \begin{bmatrix} \gamma_{\boldsymbol{i}} \boldsymbol{b}_{\boldsymbol{i}} \\ \gamma_{\boldsymbol{i}} \end{bmatrix} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta})), \qquad (11)$$

where $\boldsymbol{i}$ goes through $\mathcal{I}_2$ in the summands in both (10) and (11). $\boldsymbol{G}$ is the $3^d \times 3$ matrix of parameters of the model, in which each row i is $\boldsymbol{b}_{I_2^{-1}(i)}$.

In many applications, not only the position of the end-effector is of interest but also its orientation. The easiest way to also represent the orientation using the KB-Maps is to represent the kinematics of the unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ of the end-effector coordinate system separately in different KB-Maps. If $f : \mathbb{R}^d \to \mathbb{R}^{4\times4}$ maps joint values to the transformation matrix associated to the end-effector, the complete Bézier representation is

$$f(\boldsymbol{\theta}) \equiv \mathbb{B}(\boldsymbol{\theta}) := \begin{bmatrix} \boldsymbol{e}_1(\tau(\boldsymbol{\theta})) & \boldsymbol{e}_2(\tau(\boldsymbol{\theta})) & \boldsymbol{e}_3(\tau(\boldsymbol{\theta})) & \boldsymbol{b}(\tau(\boldsymbol{\theta})) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where $\mathbb{B} : \mathbb{R}^d \to \mathbb{R}^{4\times4}$ is the composed KB-Map, and $\boldsymbol{e}_1(\boldsymbol{\theta})$, $\boldsymbol{e}_2(\boldsymbol{\theta})$ and $\boldsymbol{e}_3(\boldsymbol{\theta})$ denote the KB-Maps of the kinematics of unit vectors.

III. Learning

Let us define a square cost function for a training set $\{(\boldsymbol{\theta}^{j)}, \boldsymbol{p}^{j)})\}_{j=1,\cdots,m}$:

$$E(\boldsymbol{G}) = \sum_j E_j(\boldsymbol{G}) = \sum_j \|\boldsymbol{f}(\boldsymbol{\theta}^{j)};\boldsymbol{G}) - \boldsymbol{p}^{j)}\|^2. \qquad (12)$$

The minimization of $E(\cdot)$ can be used to fit $\boldsymbol{f}$ to the set of training points. We can highlight the linearity of $\boldsymbol{f}$ by rewriting (10)

$$\boldsymbol{f}(\boldsymbol{\theta}^{j)};\boldsymbol{G}) = \sum_{\boldsymbol{i}} \frac{\gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}^{j)}))}{\sum_{\boldsymbol{i}} \gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}^{j)}))} \cdot \boldsymbol{b}_{\boldsymbol{i}} = \qquad (13)$$

$$\sum_{\boldsymbol{i}} \frac{\gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}^{j)}))}{\gamma^{j)}} \cdot \boldsymbol{b}_{\boldsymbol{i}} = \qquad (14)$$

$$\sum_{\boldsymbol{i}} w^{j)}_{\boldsymbol{i}} \cdot \boldsymbol{b}_{\boldsymbol{i}}, \qquad (15)$$

where $\gamma^{j)} = \sum_{\boldsymbol{i}} \gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}^{j)}))$ and $w^{j)}_{\boldsymbol{i}} = \frac{\gamma_{\boldsymbol{i}} \cdot B^2_{\boldsymbol{i}}(\tau(\boldsymbol{\theta}^{j)}))}{\gamma^{j)}}$. The quantity $\gamma^{j)}$ is common for all summands in sample j, and can be computed only once. It corresponds to the homogeneous coordinate that must be associated to $\boldsymbol{p}^{j)}$ to belong to the surface in projective space (11), hence the notation. Clearly, the selection of the best fitting parameters $\hat{\boldsymbol{G}}$ by means of the minimization of $E(\cdot)$ is a *linear* least squares problem:

$$\hat{\boldsymbol{G}} := \underset{\boldsymbol{G}}{\operatorname{argmin}}\ E(\boldsymbol{G}) = \sum_j \| \left(\sum_{\boldsymbol{i}} w^{j)}_{\boldsymbol{i}} \cdot \boldsymbol{b}_{\boldsymbol{i}} \right) - \boldsymbol{p}^{j)} \|^2. \qquad (16)$$

We can use two kinds of methods to solve this problem: exact methods and gradient methods.

Both are able to cope with irregular distributions of data in the training set, in contrast to some models like the original PSOM's that require a grid arrangement of the data. Besides, the gradient methods are naturally suited to deal with non-stationary data, a feature that is not available to PSOM's or

even to PSOM+ [8]. And since the cost function is purely quadratic, it does so without risk of failing, because there is only one global minimum.

A. Exact methods

The linear system being fitted in the least squares sense by (16) is:

$$\boldsymbol{W} \cdot \boldsymbol{G} = \boldsymbol{P}, \tag{17}$$

where $\boldsymbol{W}$ is a $m \times 3^d$ matrix composed of columns $\boldsymbol{w}^{j)} = (w^{j)}_{I_2^{-1}(1)}, \ldots, w^{j)}_{I_2^{-1}(3^d)})$ and $\boldsymbol{P}$ is an $m \times 3$ matrix in which row j is $\boldsymbol{p}^{j)}$. This system has enough data to determine a solution for $\boldsymbol{G}$ if $m \geq 3^d$. In this case, the linear least squares problem has a unique solution (if the columns of $\boldsymbol{W}$ are linearly independent) obtained by solving the normal equation:

$$(\boldsymbol{W}^t \boldsymbol{W}) \cdot \hat{\boldsymbol{G}} = \boldsymbol{W}^t \cdot \boldsymbol{P}. \tag{18}$$

$\hat{\boldsymbol{G}}$ can be determined by some standard method, such as QR-decomposition. If the data $\{(\boldsymbol{\theta}^{j)}, \boldsymbol{p}^{j)})\}_{j=1,\cdots,m}$ comes from noise-free FK, because any FK of d degrees of freedom can be expressed with $\boldsymbol{f}(\boldsymbol{\theta}; \boldsymbol{G})$, equation (17) will be satisfied exactly, i.e, $E(\hat{\boldsymbol{G}}) = 0$. Since the solution is unique, $\boldsymbol{f}(\boldsymbol{\theta}; \hat{\boldsymbol{G}})$ is the only FK function satisfying the data and, thus, the one that generated them. Consequently, generalization (both interpolation and extrapolation) will be perfect.

Of course, this happens in the absence of noise, but as it will be shown in the experimental Section IV, even with noisy data, we need a low number of samples to get a good approximation of the underlying FK.

In case there is no possibility to acquire enough data, i.e. the system of linear equations is underdetermined, it is still possible to find the solution that lies closest to an *a priori* estimate of the model (e.g. as a result of simulations). This can be done using, for instance, the Moore-Penrose pseudo inverse [12]. Finally, these exact learning techniques can be used repeatedly when some new data are acquired to generate successively improved models. Optionally, old data could be discarded when new ones are acquired, leading to an adaptive model.

B. Gradient methods

The derivative of $E_j(\boldsymbol{G})$ with respect to $\boldsymbol{b_i}$ (a row of $\boldsymbol{G}$) is easily obtained:

$$\frac{\partial E_j}{\partial \boldsymbol{b_i}} = (\boldsymbol{f}(\boldsymbol{\theta}^{j)}) - \boldsymbol{p}^{j)})\ w_i^{j)}. \tag{19}$$

This permits the application of an on-line implementation of linear regression, by updating each $\boldsymbol{b_i}$ after the presentation of a new sample $(\boldsymbol{\theta}^{j)}, \boldsymbol{p}^{j)})$:

$$\boldsymbol{b_i} \leftarrow \boldsymbol{b_i} - \mu(\boldsymbol{f}(\boldsymbol{\theta}^{j)}) - \boldsymbol{p}^{j)})\ w_i^{j)}, \tag{20}$$

where μ is the learning rate parameter. This update rule has been called Widrow-Hoff rule [Widrow & Hoff, 1960], delta rule, or LMS (Least Mean Squares) algorithm. Its application minimizes the mean squared error of the linear fit. It is a common practice to set $\mu = \mu_0 / ||\boldsymbol{w}^{j)}||^2$, $0 < \mu_0 \leqslant 1$, variation denoted as Normalized LMS.

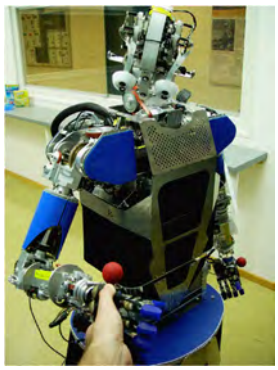
Learning by gradient methods is notoriously slower than with exact methods if a high precision is required. However, it has some advantages. The more important one is that, computationally, it is considerably lighter than exact methods. Besides, it quickly responds to dynamically changing conditions, such as easily deformable systems or the application of different tools. In general, it is naturally suited to approximate a non-stationary function.

IV. Experiments

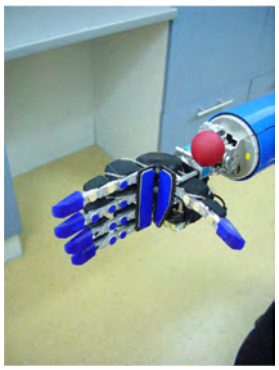
In this section, the KB-Maps presented earlier in this work are evaluated on the humanoid platform ARMAR-IIIa [1] (see Fig. 3(a)), both in experiments on the real hardware and in simulation. The ARMAR-IIIa robot contains seven independent degrees of freedom (DoF) in each arm, one in the hip and three in the head. Each arm contains a 6 DoF force sensor in its wrist. The number of joints actively used during the experiments varied, as the complexity of the learning process grows exponentially with this number. This is the reason why a smaller number was used in the experiments on real hardware than in the simulations. As our approach aims at hand-eye coordination, all experiments include joints of both the head and one arm. This way, the camera could always point in the direction of the end-effector during the experiments. On the real robot, samples were generated by manually moving the robot arm via zero-force control (see Fig. 3(a)), while an estimated FK model obtained from the geometrical model was used to fix the head looking at the hand. Joint values were then read directly from the motor encoders in order to deal with a realistic amount of sensor noise. An optical marker (a red ball signaling the end of a tool) attached to the end-effector was tracked by the built-in stereo camera system (see Fig. 3(b)), and all training samples obtained had a distance of at least 1° and maximal 3° in parameter space to their predecessors. In simulation, joint values were generated randomly in parameter space –either normally distributed or sampled through a random walk. Artificial noise was added to the positions of the end-effector in some experiments.

A. Exact Method

In the first place, simulations that show the performance of the exact learning algorithm with six DoF are presented. For training and test, two sets with 13.000 and 6.000 training experiences, respectively, and with joint angles uniformly distributed over ±80° were used. The associated positions in euclidean space were created by the FK constructed from the CAD description of the kinematics. In addition to that, an artificial noise with standard deviation σ_{noise} = 10mm was applied only to the training data. For the evaluation of the extrapolation capabilities other 6.000 independent samples from a space more than ten times larger (with angles between ±120°) were created. In the first experiment, subsets of the training data with different sizes were used



(a) Generation of training samples using zero-force control.



(b) Close-up of the optical marker attached to the right hand.

Fig. 3. The humanoid ARMAR-IIIa robot.

for learning. It is investigated how the error over the test, training and extrapolation data is related to the number of training experiences used for learning. In Fig. 4, the results of this experiment can be seen. The error on the training data (green) increases until it reaches the level of the artificial noise (grey), while the errors on the test data (blue) and the extrapolation (orange) decrease with a growing number of training data. After around 2.200 samples, the mean error on the unknown test data falls below the standard deviation of the artificial noise. Remember that the test set (unlike the training set) comes without noise, which explains why the test error becomes smaller than the training error. This means that the algorithm is capable of compensating for the sensor noise.

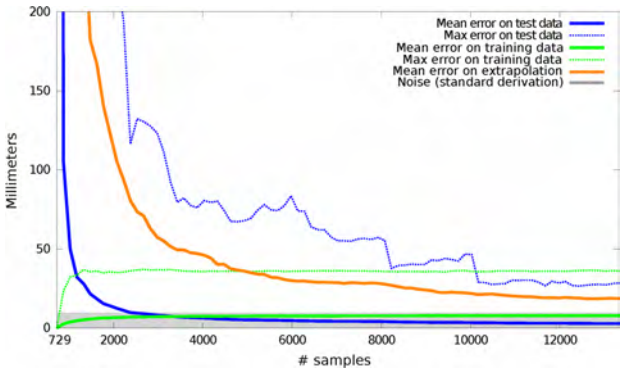


Fig. 4. Plot of a batch-learning with simulated data on the Armar-IIIa. Training samples (13.000) were generated equally distributed over $[\pm 80°]$ and with added noise of $\sigma_{err} = 10\,mm$ applied to the position $\boldsymbol{x}$ of the end-effector. A similarly generated test set with 6.000 samples and another set for extrapolation (equally distributed over $[\pm 120°]$) with 6.000 samples are included in this experiment. The figure shows the error over the learned data *(green)*, the test data *(blue)* and the extrapolation data *(orange)* in relation to a varying number of learned samples *without* online learning. Thick lines represent mean errors and dotted lines maximal errors.

Subsequently, the applicability on the real robot was examined. For this task, training experiences with five joints of the robot actively moved were produced as described in the beginning of this section. A training set with 1.000 samples and a training set with 500 samples were generated. In order to have a direct comparison, a second KB-Map was trained using exactly the same joint angles, but with the associated CAD-generated positions with an added noise of $\sigma_{noise} = 20mm$, which is approximately of the same magnitude as the one in the perception system. Fig. 5 shows the outcome of this experiment. As one can see from the similarity of both curves, the algorithm acted on real hardware as it had been predicted by the simulation.

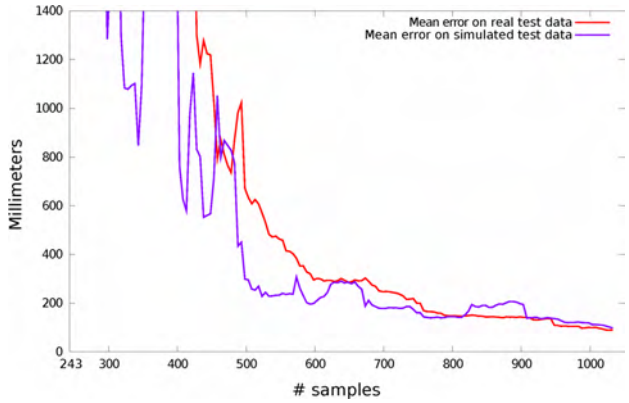


Fig. 5. Exact learning on real data recorded with 5DOF and a training set of 1000 samples and test set with 500 samples. The measured errors *(violet)* are compared to simulated values with noise $\sigma_{err} = 20\,mm$ *(red)*.

These two experiments show that the exact learning method is robust to sensor noise, and it can produce an acceptable estimation even for extrapolation points if enough training samples are provided.

B. Gradient method

In this section, the presented gradient method is integrated in a learning process that is divided into an online and an offline part. The order in which training samples are learnt is very important in the case of the gradient method. The best learning effect results from randomly generated data where consecutive samples have a larger distance in parameter space. In reality, however, this is not the case and samples that belong to the same trajectory will usually lie close to one another. This is why, in the first part of the learning process, points are learned online as they are generated. After a certain number of experiences have been acquired, these samples are randomly permuted and again learned by the net in the second stage. In this way, the accuracy of the net can be improved without the need to create new data.

As in the previous subsection, experiments were first performed on a simulated robot again. Joint values were created by a random walk in parameter space with a distance of at least 1° and a maximum of 3° between each angle vector. All joints values are normally distributed with $\sigma = 22°$ in order to create realistic trajectories. The robot now uses 7 active DoF and instead of learning the FK from scratch, this experiment simulates learning the application of a tool. Therefore, the initial KB-Map is an exact representation of the FK obtained from the CAD model. Then the training and test data were produced with a modified FK where the

TCP was moved by a distance of 250mm. In a variation of the experiment, artificial noise of $\sigma_{noise} = 20$ mm was added to the shifted TCP. The results are presented in Fig. 6. The light blue lines indicate the distance between the TCP taught positions in two consecutive learning iterations and increases as soon as the training data is permuted. As one can see, the mean error of the test data (red) and the data with artificial noise (orange) both drop very quickly. After 1.800 cycles the mean errors are about 50 mm. This shows the speed of this learning technique as well as the robustness to noise.

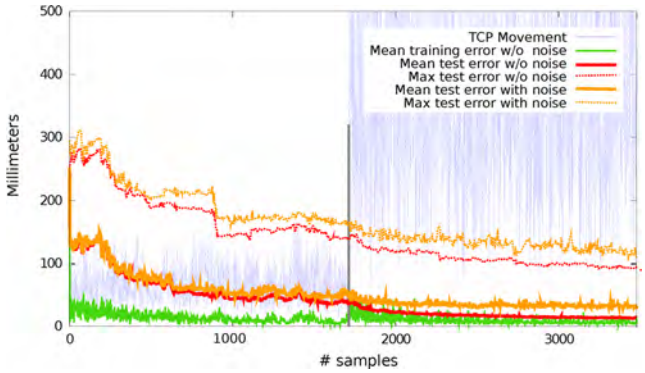


Fig. 6. Diagram of the learning progress of the incremental learning while learning a 7 DoF kinematic chain. It shows the actual error between estimation and experienced position in each iteration *(green)*, the error on the test data of learning without noise *(red)* and noise with $\sigma = 20mm$ *(orange)*. Thick lines represent mean errors and dotted lines maximal errors. The distance between the particular TCP positions is shown *(light blue)*. It increases dramatically after 1700 iterations, when the algorithm enters the second stage and the acquired experience is learnt again in random order.

In the last experiment, the learning behavior of the gradient method with simulated data and on real hardware is directly compared. The robot uses 6 active joints, and a number of 2.200 training experiences were created by moving the end-effector as described earlier in this section. The same joint values of these experiences were used in simulation to generate training samples. The outcome of the comparison of the two KB-Maps can be seen in Fig. 7. From the similarity of the two curves, it follows that the learning on real hardware succeeds as predicted by the simulation.

As a consequence from these two experiments, it can be seen that the gradient-based learning can be used to refine a crude FK model very rapidly. Thus, the obtained results proved the robustness of this learning to noise, as well as its applicability to a real humanoid robot.

V. Conclusions

In the present paper, a novel approach for learning the FK mapping as part of the body schema of humanoid robots was presented. Inspired by PSOMs, we wanted to overcome the large number of robot movements required to get a good approximation of FK.

First, since FK is a composition of circles, models based on polynomials (as PSOM) cannot exactly represent a FK. Thus, we have chosen a model based on rational Bézier polynomials –the Kinematic Bézier Maps–, which are a

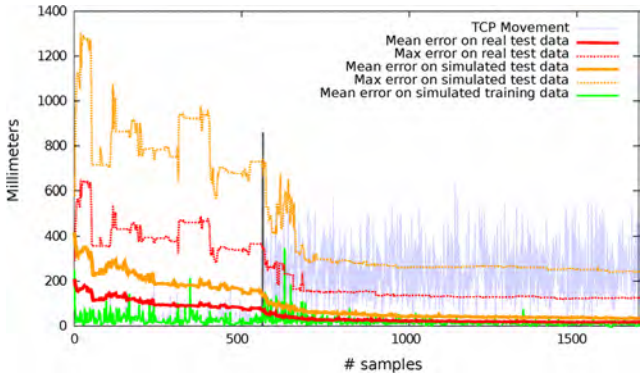


Fig. 7. Comparison of the incremental learning progress when processing real data from marker tracking (orange) and simulation w/o noise (red). Thick lines represent mean errors and dotted lines maximal errors. After 600 training experiences no new data is acquired but previously learnt data is processed anew in arbitrary order. The difference between the TCP positions is shown (light blue).

family of functions that includes the description of any FK. Besides, these functions have an important advantage: adjusting the model to a set of a sample points is a linear least squares problem.

Second, we have introduced *a priori* knowledge of the function to be learnt in the model which is the key to reducing the number of samples. This has been achieved by restricting the model to represent only compositions of a certain family of ellipses which always includes the circle. The constraints implied by this restriction are easily integrated in the linear least square problem. The approach can be summarized as reformulating the problem in a larger space –the positions of the Bézier control points–, where it becomes linearly solvable.

This higher-dimensional problem can be easily solved with any standard linear least-squares method, yielding our exact learning method. Alternatively, the least squares cost has a simple derivative, encouraging alternative algorithms, the so-called gradient learning methods, which are well suited for online-learning. Using the exact method, in the absence of noise, it is possible to learn *exactly* a FK with only 3^d samples, where d is the number of DoF, which none of the previous works was able to accomplish. And so, with an arbitrary sample distribution. This means that, even if samples are grouped in a very reduced zone of the workspace, interpolation and extrapolation are perfect.

We have carried out experiments, both simulated and in real hardware, with a humanoid robot under noisy conditions, proving that our algorithms are able to quickly learn a good approximation of the kinematics of the robot from inaccurate measures.

Our learning algorithm performs very well if enough noisy samples from the whole workspace are provided. Even if the noisy samples are restricted to a local zone of the workspace, we obtain good interpolation and extrapolation, although the last one requires more samples. But, if the samples are noisy, few and local, the algorithm performs poorly, especially in extrapolation, where it can exhibit very large errors. This

is due to the fact that with noise and scarce data, the isoparametric curves of the model become often strongly elliptical.

This provides an idea about how to improve our system in these conditions, although there do not exist any easy solution because the constraints to enforce complete circularity are non-linear. Finally, a less challenging future work is to deal not only with rotational joints, but to generalize the model for robots having any combination of prismatic and rotational joints.

APPENDIX

A. Isoparametric Curves of the Multidimensional Model

A d-dimensional tensor product Bézier form of degree 2 in which the vector $\boldsymbol{i}$ is spelled out for convenience, has the form:

$$\mathbb{b}(s_1,\ldots,s_d) = \sum_{i_1,\ldots,i_d=0}^{2} \mathbb{b}_{i_1,\ldots,i_d} \cdot B^2_{i_1,\ldots,i_d}(s_1,\ldots,s_d). \quad (21)$$

Without loss of generality, we show the isoparametric curve of this Bézier form when s_1 is the free variable. The above equation can be rewritten as:

$$\sum_{k=0}^{2} B^2_k(s_1)\Big(\sum_{i_2,\ldots,i_d=0}^{2} B^2_{i_2,\ldots,i_d}(s_2,\ldots,s_d)\cdot \mathbb{b}_{k,i_2,\ldots,i_d}\Big). \quad (22)$$

We can define a new function $\mathfrak{q}_k(s_2,\ldots,s_d)$ to rename the expression in the big parenthesis; when $s_2,\ldots,s_d$ are fixed, $\mathfrak{q}_k$ is a constant and (22) becomes a single-variable Bézier curve defined by the control points $\mathfrak{q}_0$, $\mathfrak{q}_1$ and $\mathfrak{q}_2$:

$$\sum_{k=0}^{2} B^2_k(s_1)\cdot \mathfrak{q}_k(s_2,\ldots,s_d). \quad (23)$$

Let the homogeneous coordinates of $\mathfrak{q}_0$, $\mathfrak{q}_1$ and $\mathfrak{q}_2$ be ϖ_0, ϖ_1 and ϖ_2, respectively. To be an ellipse, ϖ_0=ϖ_2 and $\varpi_1/\varpi_0 < 1$ must be satisfied. Remind that we set the weights $\gamma_{i_1,i_2,\ldots,i_d}$ of control points $\mathbb{b}_{i_1,\ldots,i_d}$ to $\gamma^{ones(i_1,\ldots,i_d)}$, where $ones()$ returns the number of ones in the arguments and γ is an arbitrary constant minor than one.

The values of the ϖ's are then

$$\varpi_0 = \sum_{i_2,\ldots,i_d=0}^{2} B^2_{i_2,\ldots,i_d}(s_2,\ldots,s_d)\cdot \gamma_{0,i_2,\ldots,i_d}$$

$$\varpi_1 = \sum_{i_2,\ldots,i_d=0}^{2} B^2_{i_2,\ldots,i_d}(s_2,\ldots,s_d)\cdot \gamma_{1,i_2,\ldots,i_d}$$

$$\varpi_2 = \sum_{i_2,\ldots,i_d=0}^{2} B^2_{i_2,\ldots,i_d}(s_2,\ldots,s_d)\cdot \gamma_{2,i_2,\ldots,i_d}$$

Everything in the development of ϖ_0 is the same as that in ϖ_2, except the first index in the weights, which is 0 for ϖ_0 and 2 for ϖ_2. Since $\gamma_{0,i_2,\ldots,i_d} = \gamma^{ones(i_2,\ldots,i_d)}$ and $\gamma_{2,i_2,\ldots,i_d} = \gamma^{ones(i_2,\ldots,i_d)}$, we conclude that ϖ_0=ϖ_2. Similarly, ϖ_0 and ϖ_1 differ only in the first index of all involved weights. Those in ϖ_1 are $\gamma_{1,i_2,\ldots,i_d} = \gamma^{ones(i_2,\ldots,i_d)+1}$, which means that they correspond to those involved in ϖ_0 multiplied by γ. Therefore, the conditions $\varpi_1 = \varpi_0\gamma$ and $\varpi_1/\varpi_0 = \gamma < 1$ are met, which concludes the proof that, with the chosen weights for control points $\mathbb{b}_{i_1,\ldots,i_d}$, the isoparametric curves of (21) are ellipses.

REFERENCES

[1] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. Armar-iii: An integrated humanoid platform for sensory-motor control. pages 169–175, Dec. 2006.

[2] V. R. de Angulo and C. Torras. Speeding up the learning of robot kinematics through function decomposition. *IEEE Transactions on Neural Networks*, 16(6):1504–1512, 2005.

[3] V. R. de Angulo and C. Torras. Learning inverse kinematics: Reduced sampling through decomposition into virtual robots. *IEEE Transactions on Systems, Man and Cybernetics - part B*, 38(6):1571–1577, 2008.

[4] A. D'Souza, S. Vijayakumar, and S. Schaal. learning inverse kinematics. In *ieee international conference on intelligent robots and systems (iros 2001)*. piscataway, nj: ieee, 2001.

[5] G. E. Farin. *NURBS: From Projective Geometry to Practical Use*. A. K. Peters, Ltd., Natick, MA, USA, 1999.

[6] G. E. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[7] M. Hersch, E. Sauser, and A. Billard. Online learning of the body schema. *International Journal of Humanoid Robotics*, 5(2):161–181, 2008.

[8] S. Klanke and H. J. Ritter. Psom+ : Parametrized self-organizing maps for noisy and incomplete data. In *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM 05)*, Paris, France, Sep 2005.

[9] A. Maravita and A. Iriki. Tools for the body (schema). *Trends in Cognitive Sciences*, 8(2):79 – 86, 2004.

[10] A. Maravita, C. Spence, and J. Driver. Multisensory integration and the body schema: Close to hand and within reach. *Current Biology*, 13:531–539, 2003.

[11] W. Penfield and T. Rasmussen. *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*. Macmillan, 1950.

[12] R. Penrose. A generalized inverse for matrices. In *The Cambridge Philosophical Society*, 51, pages 406–413, 1955.

[13] H. Prautzsch, W. Boehm, and M. Paluszny. *Bezier and B-Spline Techniques*. Springer-Press New York, Inc., Secaucus, NJ, USA, 2002.

[14] M. I. Stamenov. Body schema, body image, and mirror neurons. In H. D. Preester and V. Knockaert, editors, *Body Image and Body Schema*, pages 21–43. De Preester, 2005.

[15] J. A. Walter. PSOM network: Learning with few examples. In *In Proc. Int. Conf. on Robotics and Automation (ICRA-98*, pages 2054–2059, 1998.

[16] D. Whitney. Resolved motion rate control of manipulators and human prostheses. *Man-Machine Systems, IEEE Transactions on*, 10(2):47–53, June 1969.