

Project no.: 027657

Project full title: Perception, Action & Cognition through learning of Object-Action Complexes

Project Acronym: PACO-PLUS

Deliverable no.: D4.1.3

Title of the deliverable: Technical report on generalization of affordances across objects

Contractual Date of Delivery to the CEC:	31 January 2009
Actual Date of Delivery to the CEC:	26/2-2009
Organisation name of lead contractor for this deliverable:	SDU
Author(s):	Norbert Krüger, Florentin Wörgötter, Tamim Asfour, Rüdiger Dillmann, Justus Piater, Mark Steedman, Aleš Ude, Alejandro Agostini, Danica Kragic, Jan-Olof Eklundh, Bernhard Hommel, Dirk Kraft, and Renaud Detry
Participant(s):	BCCN, KTH, JSI, UniKarl, CSIC, UEDIN, UL
Work package contributing to the deliverable:	WP1, WP2, WP4.3, WP4.1, WP5.2
Nature:	R
Version:	Draft
Total number of pages:	18
Start date of project:	1 st Feb. 2006 Duration: 48 month

**Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)
Dissemination Level**

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This technical report addresses the work the PACO consortium has done in WP4.1 in month 25–36. We will describe work on intermediate representations and actions associated to these. In particular, we describe processes which concern learning of actions associated to specific objects as well as cross object generalization. This deliverable covers 5 journal publications [D, K, L, H, G] (some of them in the status of being submitted) and 8 conference or workshop publications [C, F, E, I, M, J, A, B].

Keyword list: OACs, Structural relations, Grasping, Pushing and Filling

Table of Contents

1. INTRODUCTION	3
2. OBJECT AND SCENE REPRESENTATIONS	5
2.1 STRUCTURAL RELATIONS ON HIGHER LEVEL ENTITIES	5
2.1.1 <i>Contour Relations in 3D</i>	5
2.1.2 <i>Storing temporally disambiguated Surface Knowledge in Graphs</i>	7
2.2 OBJECT LEARNING AND POSE ESTIMATION	8
2.3 MULTI VIEW REPRESENTATIONS	9
2.3.1 <i>Multi-View Object Representations in Visual Search</i>	9
3. ACTIONS	10
3.1 GRASPING WITHOUT OBJECT KNOWLEDGE	10
3.1.1 <i>Grasping behaviour based on Edge Information</i>	10
3.1.2 <i>Grasping behaviour based on Surface Information</i>	11
3.2 GRASPING WITH PRIOR OBJECT KNOWLEDGE	11
3.3 PUSHING	12
3.4 FILLING	13
3.5 MULTI CUE–AFFORDANCE RELATIONS	14
3.6 RULE SYSTEM - ACTIONS AFFORDANCE	15
4. CONCLUSION	17

1. Introduction

Affordances can be highly object dependent, as for example the learning and storage of successful grasps of specific objects. In this case, generalization will improve the ability to perform actions on those specific objects. Cognitive systems are able to generalize across objects based on the experiences made on such specific objects. Both kinds of generalization, object specific and cross-object generalization, can be performed by cycles of execution and learning as formulated in the OAC concept. In this deliverable, we describe the representations on which such generalisation methods are based on in section 2 and we describe work on the associated actions in section 3. The work described here have to be seen as sub-modules that become combined in a cognitive architecture developed in the other WPs, mainly WP1.2, WP4.2 and WP4.3 (see also Deliverable D1.2.2 and D4.3.5).

Our visual representations (already introduced in former deliverables, see D4.1.1 and D4.1.3) provide rich and structured information on which generalisation processes takes place. In the last year, we extended our work on 3D structural relations. In this context, we encountered the problem that for a workable definition of 3D relations the uncertainty in the reconstruction process needs to be taken more closely into account (see section 2.1.1). This allowed for a significant progress in three aspects: First, we could stabilize the grasping behaviour by having better estimates of the induced actions (see section 3.1.1). Second, it allowed for an extension of the generic grasping behaviour to a part-based approach as outlined in section 3.1.1 and [C]. In addition and most importantly, it allowed for a representation of objects and scenes by their structural relations in histograms of high level attributes and relations. In these histograms, structural similarities are preserved in similarities of histogram intersections (see section 2.1.1). We anticipate, that these histograms give way to the learning of part-action associations which we will pursue in the last year of the project.

Moreover, to apply previously stored knowledge that can be associated to an object, it is required to find the pose of the object based on a learned representation. In the last year, we have developed a pose estimation algorithm based on the autonomously extracted visual representations based on edge information (see section 2.2, [F] and [I]). As a complementary mechanism, we describe in section 2.3 an approach which supports the generation of hypotheses of object locations based on appearance-based visual representation. The approach comprises visual search, attention and active guidance of the gaze of the Karlsruhe Humanoid Head. While 3D information is crucial for the generalisation of affordances, the appearance-based part of the representation assures a robust visual perception and provides mechanisms to restrict the structural analysis of the scene to only salient regions (see also [M]).

We have also extended the edge based representations by surface based representations from which high level features in terms of surface areas (as well as their underlying 3D structure) with their spatial relations become stored in a graph (as described in section 2.1.2 and [D]). In this, a certain stability is induced by the continuity of the action (in this case filling). Moreover, the representation allows for efficient comparison of different instances of the same or different actions which we will address in the final year.

Based on the representations described in section 2, we can now associate actions to those. The actions we are dealing with are positioned at different levels of the hierarchy and presuppose different levels of prior knowledge as described in section 3. We refer to grasping (section 3.2 and 3.1) and non-prehensile actions such as pushing (section 3.3) as well as high level action such as filling (section 3.4) and more general object affordance relations (see section 3.5). Finally, we address rule-learning (section 3.6) where action rules are encoded at the higher level of the architecture using symbolic descriptions of more generic actions like grasping or pushing. Rules refer to the actions to be performed to produce changes in the object but not to how they become performed, which is the task of lower levels of the architecture.

In the last year, we have made use of the structural relations provided by the visual representations (as discussed in section 2) to extend the generic grasping behaviour not requiring any specific object knowledge by learning (see section 3.1.1 and [K]). As a complementary strategy, surface information is used to trigger

another grasping behaviour not requiring specific object knowledge based on the decomposition of objects into 3D boxes (see section 3.1.2 and [I]).

An important step for the development of the final system has been the learning of grasping using prior object knowledge which is described in section 3.2 and [E]. Based on the concept of grasp densities, we are able to store grasping experiences in an efficient way. Although here the actual affordance is object dependent, we expect for the final year to be able to extract more generic knowledge by learning visual parts that share similar grasp properties across different objects. This way, a grasp affordance will be directly and exclusively connected to the visual evidence that predicts its applicability, allowing for its generalization across objects.

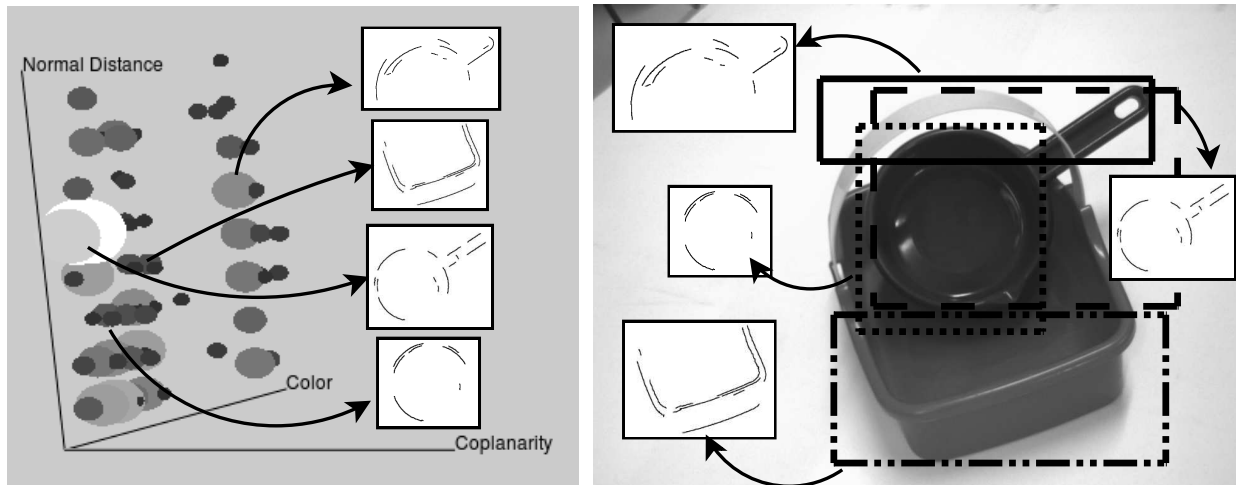


Figure 1: A sample 3D histogram of contour relations and the contours that created the certain bins of the histogram. Note that the spheres represent the locations of histogram bins and the brightness and diameter of these spheres are directly proportional to the value stored in the bins (a crowded bin is represented as a big and brighter sphere). **(a)** 3D histogram. **(b)** Parts of the scene that created the bins shown in figure (a).

The learning of 'pushing' as an example of a non-prehensive action is addressed in section 3.3 and [J]. Although right now the algorithm is object dependent, we are currently extending the algorithm such that it can be applied to arbitrary objects. Work on the learning of general object–action relations for the action 'filling' is described in section 3.4. In section 3.5 and [L], a generalization of object affordances

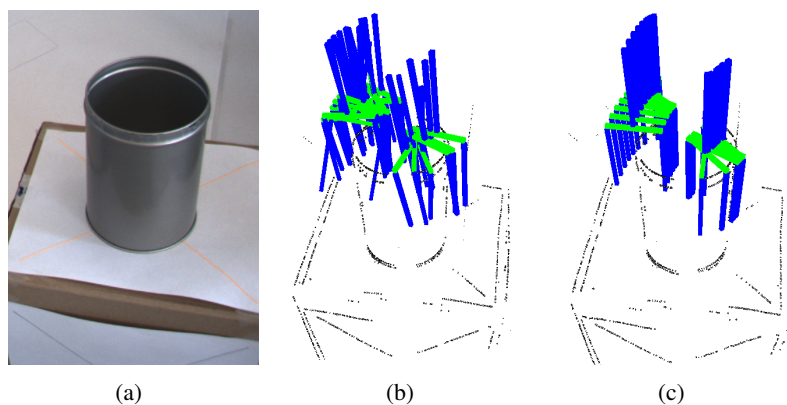


Figure 2: Grasp locations for different approaches. **(a)** Object to be grasped. **(b)** Grasps that are calculated by using local features for different noise levels. **(c)** Grasps that are calculated by using contours for different noise levels

looking beyond the actions connectable to the interacting system (the robot hand) is outlined. In addition, we consider actions in the sense of the perceptual system and processing, since different objects may afford different processing techniques. As a main example, uniformly colored objects are well suited for the edge-based grasp hypotheses generation from visual data (section 3.1.1), while textured objects are for the surface-based grasp hypotheses generation (section 3.1.2).

Learning on the highest level of the processing hierarchy is described in section 3.6 and [A, B]. In this section, a learning method is presented where the affordance of abstract action rules is learned: The system learns the relevant attributes to successfully obtain changes over an object after the execution of an action. The actions are described symbolically and reference one or more generic actions learned at lower layers of the architecture. Action rules can encode any object-action affordance, as far as the objects and actions are described symbolically.

This deliverable covers 5 journal publications [D, K, L, H, G] (some of them in the status of being submitted) and 8 conference or workshop publications [C, F, E, I, M, J, A, B].

2. Object and Scene Representations

In this section, we describe work on object and scene representation in the context of WP4.1.

2.1 Structural Relations on Higher Level Entities

Our object representations contain rich structures information covering 2D and 3D information. This structure linked to actions and is used in the learning and generalization processes. In section 2.1.1, we describe work on contour relations while in section 2.1.2, we describe work on surface relations.





















2.1.1 Contour Relations in 3D

We have established a vision system in which contour relations (such as coplanarity, cocolority and distance) have been computed (see D4.2). The aim is to code objects not only by their appearance but also by the structural relations between higher level entities they are consisting of. Besides extending and stabilizing these relations, we have established a method to code objects and scenes by means of histograms of these relations. Figure 1 shows a 3D histograms of a sample scene. As shown in the figure, certain histogram bins corresponds to specific object parts and this leads naturally to the learning of parts as higher level constallations occuring frequently across objects.

We also computed the histogram intersections for different objects as an object identification task. As shown in Table 1 we achieve a high robustness towards view changes and more importantly we can show that structural similarities in the objects lead to high similarities on the histogram intersection (e.g., the similarity between plate and pan).

An important issue while reasoning in 3D is to take into account the uncertainty of the reconstruction process. In Figure 2, sample grasps for different random noise levels are shown. In Figure 2 (b), grasps are calculated by using the local approach. The grasps that are calculated by using contours take into account the uncertainty of the data for the same noise levels are shown in Figure 2 (c). As we see in the figure, even though the best grasps have been chosen for the local approach, the global approach performs significantly better.

Table 1: Histogram intersections for different objects with different poses.

										
	0.3008	0.0200	0.0290	0.2798	0.0347	0.1662	0.0920	0.0863	0.0280	0.1134
	0.0041	0.2365	0.0383	0.0282	0.0051	0.1144	0.1088	0.0998	0.0000	0.0000
	0.0021	0.0244	0.1449	0.0030	0.0051	0.0404	0.0000	0.0744	0.0495	0.0361
	0.2219	0.0255	0.0001	0.4903	0.0081	0.0740	0.0534	0.0438	0.0022	0.1863
	0.0240	0.0000	0.1262	0.0036	0.2935	0.0100	0.0038	0.0075	0.1671	0.1309
	0.0233	0.1595	0.0234	0.0548	0.0102	0.1984	0.1097	0.0998	0.0000	0.0234
	0.0144	0.1357	0.0000	0.0213	0.0000	0.0848	0.2008	0.1026	0.0000	0.0267
	0.0230	0.0085	0.0836	0.0204	0.0341	0.0295	0.0085	0.1239	0.0218	0.0870
	0.0445	0.0077	0.0914	0.0144	0.1023	0.0302	0.0038	0.0462	0.3085	0.2278
	0.1541	0.0026	0.1719	0.1110	0.1683	0.0510	0.0573	0.0935	0.2670	0.4089

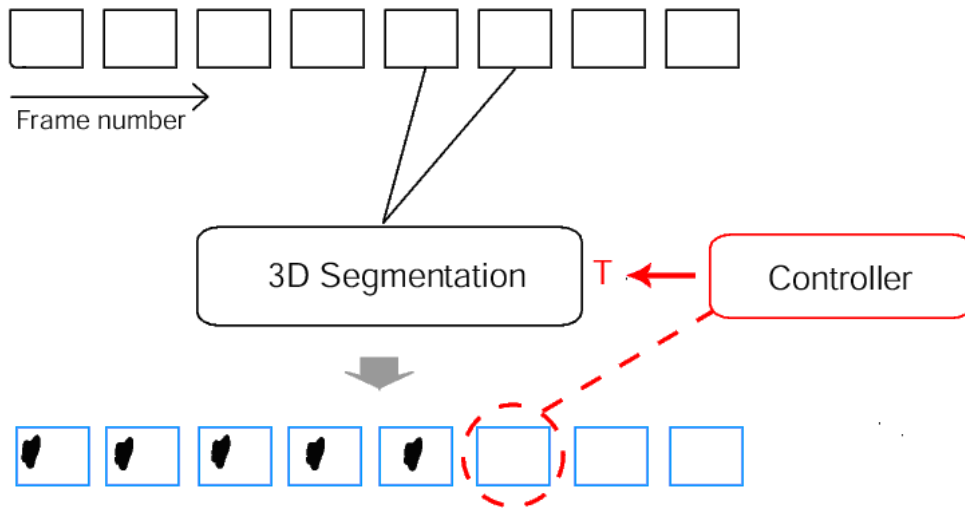


Figure 3: A schematic of the entire system. The feedback controller detects the instable segmentations and rectifies them by adjusting temperature value of the core algorithm.

2.1.2 Storing temporally disambiguated Surface Knowledge in Graphs

Our main aim is to represent visual scenes with semantic graphs to achieve object tracking. We present an algorithm for segment tracking based on a novel, conjoint framework, combining local correspondences and image segmentation to synchronize the segmentation of subsequent images in a movie. The main idea behind the algorithm is to provide a partitioning of the image sequence in segments, such that points in a segment are more similar to each other than to points in another segment, and such that corresponding image points belong to the same segment [D]. In the algorithm the segmentation process of the images is based on the method of superparamagnetic clustering. In this method each image pixel is represented by a Potts model of spins which can have different energy states. Neighboring spins interact such that spins corresponding to pixel of similar gray values tend to be in the same spin state (see, e.g., [2]). Spin interactions result in the formation of clusters of correlated spins, providing an automatic labeling of corresponding image regions. In the case of the application of the clustering method to image sequences, we split the image sequences into pairs where the last frame of the previous pair is identical to the first frame of the next pair. The relaxation results (spin states) are transferred across image pairs, such that the spin states of the previous image pair are assigned becoming the initial spin states for the next image pair. Since the last frame of the first pair is identical to the first frame of the next pair, we assign the same segment labels to the segments which cover similar image regions. Therefore, segment tracking can be achieved for the whole image sequence. However, segment tracking might fail due to the light reflections or similar changes in the scene because the segmentation process of an image is sensitive to global and local changes of the image. As a solution, we present a feedback control mechanism which detects the segment instabilities and rectifies them by adjusting temperature value of the core algorithm [D]. The feedback control mechanism is based on the size information of the segments and assumes that "good" segments change their size in a continuous "predictable" manner. Thereby, the sudden changes of the segment sizes, e.g. merging of two neighbor segments due to insufficient illumination, can be detected and corrected at higher temperature by the control mechanism. Since the temperature choice affects the formation of segments, it is a crucial element for the controller unit. A schematic of the entire system, i.e. core algorithm with feedback control, is shown in Fig. 3. Segments are then represented by graphs in which the nodes are segment labels plotted at the center of segments and the edges point out the neighborhood relationships between the segments. An example of continuous segment tracking for real movies with graph representations is depicted in Fig. 4. As future work we aim to track segments in a more complex scenario with OACs. Additionally, we aim to charge

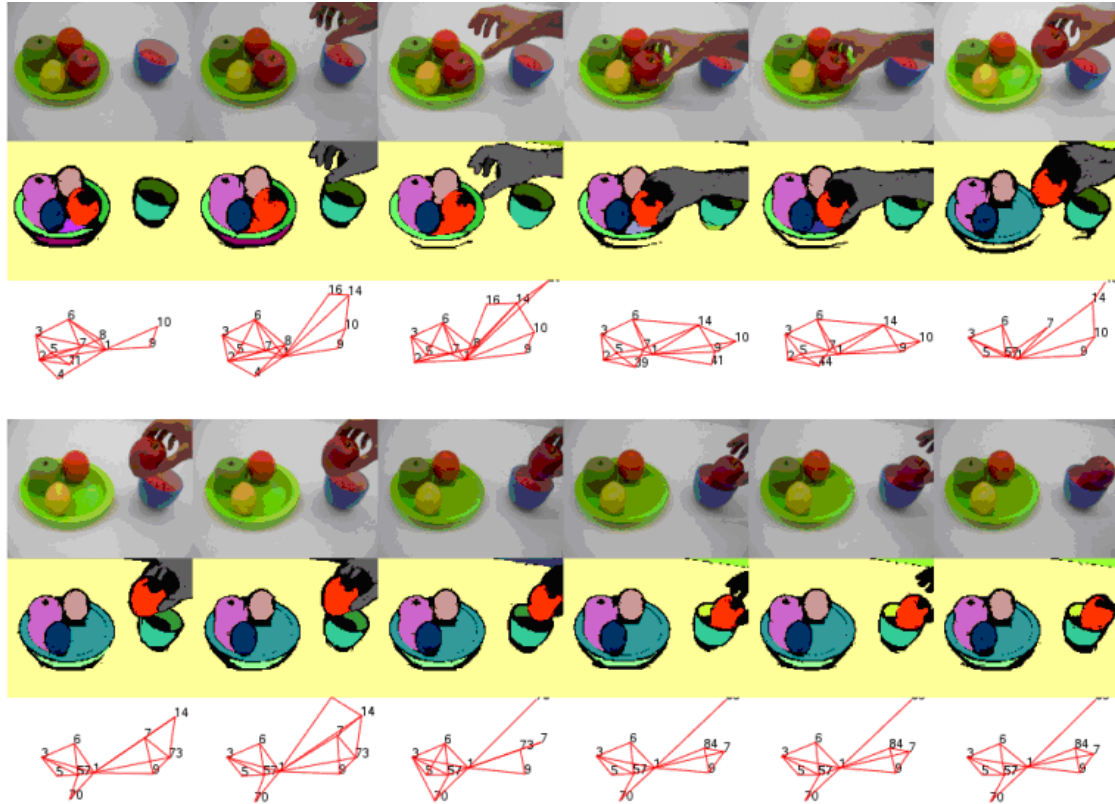


Figure 4: Continuous segment tracking for real movies with graph representations. The algorithm with feedback control mechanism is applied to a movie showing a hand replacing a red apple in a scene. The segments are represented by graphs in which the nodes are segment labels plotted at the center of segments and the edges point out the neighborhood relationships between the segments. The red apple indicated by node number 7 can be tracked continuously during the whole image sequence.

the nodes with some extra meaningful values such as 3D chain codes of the object boundaries and planar patch information of the object surfaces. Rotation, scaling and translation invariant chain codes and planar patches can help us in order to recognize the objects in 3D space while tracing the segments 2D space. For this purpose we use stereo vision to perceive the depth information of the objects. In the mean time, we are working on a parallel implementation of the algorithm on GPUs to achieve real-time segment tracking for robot applications.

2.2 Object Learning and Pose Estimation

In our work, we organize local visual data such as SDU edges (or potentially BCCN surfaces) into object models that allow for pose recovery in cluttered scenes. The probabilistic nature of our object representation and detection algorithms provides intrinsic robustness to noise and perceptual uncertainty. Although we currently focus on the modeling of specific objects, our models can manage small morphological variations implicitly, treating discrepancy as input noise. We can also explicitly learn small variations, such as the freedom of an articulated object part, by presenting the learning algorithm with different object-part configurations. For example, Figure 5 shows a weighing-scale model aligned to two different configurations of the scale. The model was learned from SDU edges extracted from 5 images showing the scale at varying equilibrium positions. This work is further described in [F] and [1].

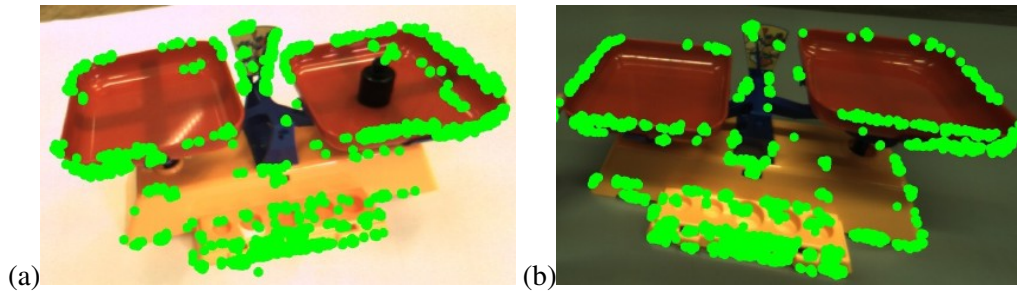


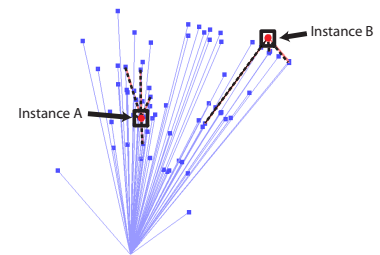
Figure 5: Object model fitted to two different articulated configurations.



(a) Scene setup used for the complex search task. Two instances of one object are presented to the system in a cluttered scene.



(b) Resulting saliency sphere and foveal views of the left foveal camera. In the final state, the system focusses alternately on the position of both object instances.



(c) Content of the scene memory after 29 saccadic eye movements. Each verified instance is supported by several hypotheses from the coarse object search procedure.

Figure 6: Results of the object search task for two instances in a complex scene.

2.3 Multi View Representations

The goal of the work done at UniKarl is the development of object representations, which supports separating between objects and the generation of hypotheses of object locations using on appearance-based multi-view visual representation as well as the application of such representations for visual search and object separation task on a humanoid robot. In previous experiments we investigated how the ability of separating between objects can benefit from multi-view representations on a robot platform that is able to actively control the movement of the object [4]. The approach allows to separate between ambiguous objects by revealing views that are best suited to distinguish between the spurious hypotheses. This approach has been extended toward the generation of hypotheses during a visual search task in cluttered scenes by actively guiding the gaze of the Karlsruhe Humanoid Head.

2.3.1 Multi-View Object Representations in Visual Search

The object search process is a common daily human activity. Almost all actions that humans perform rely on specific items which support the action e.g. as tools. In [M], we propose an approach which performs a visual object search task based on the developed multi-view object representation scheme (see [5]). Instead of successively filtering the visual stimuli starting with low-level cues as in traditional attention systems [6], our approach starts with a search for the object in the scene with coarse features. Using the resulting matches, we follow a hypothesis and test procedure in order to verify the matches with local, more descriptive features. In order to store the object information from the current scene as collected during the object search process, a scene memory based on the Sensory Ego Sphere based approach (see [3]) is proposed which ensures the persistence and consistence of already acquired information about the scene. The scene memory allows for

the integration of multiple hypotheses based on spatial coherence, which makes the search task more robust. The gaze of the Karlsruhe Humanoid Head used in the experiments is directed based on the content of the scene memory using spherical saliency maps. Experiments comprising the search for one object at a time and the search for multiple instances of an object in cluttered scenes were carried out. Fig. 6 shows the results of a visual search task. The system performed several saccadic eye movements initialized by the attention component. Each saccade results in a verification of possible locations for the searched object in the scene. The scene memory is successively updated and finally contains the locations of the two instances of the searched object. In this state, the system focuses on both instances alternately. The content of the scene memory provides the basis for further tasks such as grasp hypotheses generation and active object separation.

3. Actions

In the following subsections and based on the representations described in section 2, we deal with a number of actions being addressed in PACOplus as well as the learning and generalization processes being involved.

3.1 Grasping without Object Knowledge

In this subsection, we address two complementary grasping mechanisms not requiring object-specific knowledge based on edge and surface information.

3.1.1 Grasping behaviour based on Edge Information

Grasping based on Co-planarity: In the work [K], we describe and evaluate a grasping mechanism that does not make use of any specific object prior knowledge. The mechanism makes use of second-order relations between visually extracted multi-modal 3D features that become provided by an early cognitive vision system. More specifically, the algorithm is based on two relations covering geometric information in terms of a co-planarity constraint as well as appearance based information in terms of co-occurrence of colour properties. We show that our algorithm, although making use of such rather simple constraints, is able to grasp objects with a reasonable success rate in rather complex environments (i.e., cluttered scenes with multiple objects).

Moreover, we have embedded the algorithm within a cognitive system that allows for autonomous exploration and learning in different contexts. First, the system is able to perform long action sequences which, although the grasping attempts not being always successful, can recover from mistakes and more importantly, is able to evaluate the success of the grasps autonomously by haptic feedback (i.e., a force torque sensor at the wrist and information about the distance of the gripper after a grasping attempt). Such labeled data is then used for improving the initially hard-wired algorithm by learning. Moreover, the grasping behaviour has been used to trigger higher level processes such as object learning and learning of object specific grasping.

Grasping based on Parts: In the work [C], we address the problem of 3D circle detection in a hierarchical representation which contains 2D and 3D information in the form of multi-modal primitives and their perceptual organizations in terms of contours. Semantic reasoning on higher levels leads to hypotheses that then become verified on lower levels by feedback mechanisms. The effects of uncertainties in visually extracted 3D information can be minimized by detecting a shape in 2D and calculating its dimensions and location in 3D. Therefore, we use the fact that the perspective projection of a circle on the image plane is an ellipse and we create 3D circle hypotheses from 2D ellipses and the planes that they lie on. Afterwards,

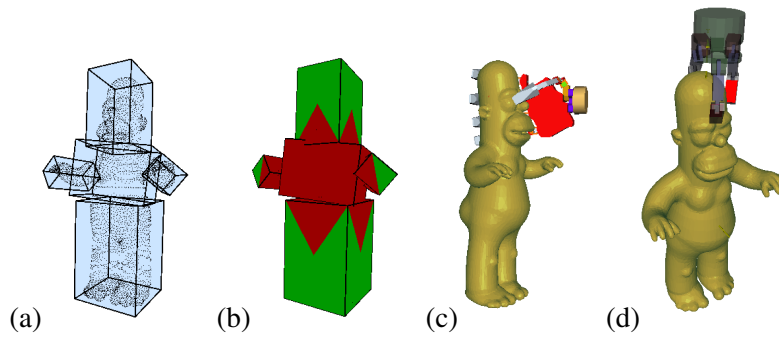


Figure 7: (a) Approximation of a simulated model’s 3D point cloud. (b) Visualization of heuristic hypothesis reduction with valid (green) and invalid (red) regions. (c) Best grasp after simulated grasp-contact learning using a hand with unknown kinematics, but a given grasp pre-shape. (d) Best grasp after simulated grasp-contact learning using known kinematics and finger positioning.

these hypotheses are verified in 2D, where the orientation and location information is more reliable than in 3D. For evaluation purposes, the algorithm is applied in a robotics application for grasping cylindrical objects.

3.1.2 Grasping behaviour based on Surface Information

Using certain higher-level object features such as shape, size, or whether the object is textured or smooth, specific objects can be mapped to certain actions and model this at the symbolic level by examining how objects, actions, and the effects of those actions relate to each other. However, different object appearance also affords different methods to encounter such meaningful features. The previous case of grasping behaviour based on 3D edge information is mainly supported by uniformly colored objects. On the other hand, textured objects enable surface information from disparity and dense 3D point clouds.

Considering textured objects and dense 3D data, it has been observed in the literature that the approximation of such data by shape primitives, e.g. spheres, boxes or cones, is a very valuable step. In this context, a focus on a simple and efficient box approximation technique has further proven to be meaningful for connecting such shape information with pre-grasp configurations, as proposed in [I]. The output of such an approximation can be interpreted as a part-description of an object, while its geometric simplicity - a constellation of boxes - not only allows for a tremendous reduction of possible grasp configurations on the object through heuristic reasoning. It also enables the connection of such representations to successful grasps by learning of contact-level grasp qualities in the force domain.

Simulation is broadly used as a helpful tool for learning and evaluating grasps. However, even in this context, the system embodiment itself may afford different strategies to define the learning. For example, if the kinematics of the gripper are unknown or a grasp pre-shape classification (i.e. power grasp, pinch grasp, etc.) is available, learning may focus on the representation itself [I].

3.2 Grasping with prior Object Knowledge

We memorize knowledge about grasp affordances in *grasp densities* attached to a mid-level visual model [E]. We use the term *grasp density* to refer to a continuous, probabilistic representation of an object grasp affordance. Through grasp densities, we organize and store the whole knowledge that an agent has about the grasping of an object, in order to facilitate reasoning on grasping solutions and their likelihood of success.

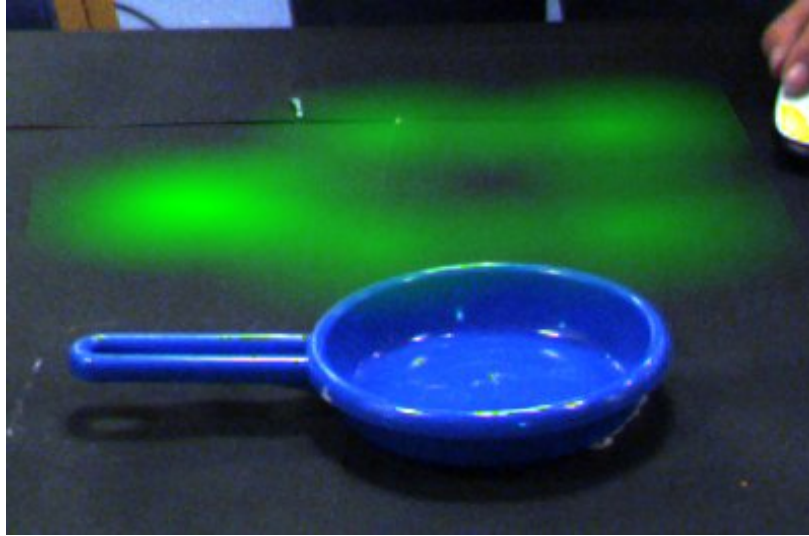


Figure 8: Visualization of the gripper position part of a density for a pinch grasp on a toy pan. Brighter green areas indicate where a gripper should be placed to successfully pinch-grasp the object. The orientation part of the density is not rendered in this illustration.

We represent the affordance of an object for a given grasp preshape through a continuous probability density function defined on the 6D gripper pose space, within an object-relative reference frame (see Fig. 8). Grasp densities are initially learned from grasps computed from visual cues (see previous section), using automatic learning techniques to turn sets of discreet grasps into continuous *grasp proposal densities*. These densities are attached to the visual object model mentioned above [E], which allows a robotic agent to execute *samples* from a grasp proposal density under arbitrary object poses. Observing the outcomes of these grasps allows us to learn from experience: we apply machine learning algorithms on grasp outcomes and learn *grasp empirical densities*, which form a finer representation of object properties.

The visual object representation considered here organizes object parts in a hierarchy of features [E]. After grasp affordance have been attached to the model, some features correspond to visual representations of parts of the objects, other features relate to grasp affordances. We currently learn visual and grasp features independently, and connect them through a single top-level model feature. One of our goals is to learn visual parts that share the same grasp properties across different objects. This way, a grasp feature will be directly and exclusively connected to the visual evidence that predicts its applicability, allowing for its generalization across objects.

3.3 Pushing

The goal of the research in [J] is to investigate how to acquire useful action knowledge by observing the results of exploratory movements on objects that afford a certain action. We focus on poking as a representative type of nonprehensile manipulation. Poking can be defined as a short term pushing action. We proposed an explorative process that enables the robot to learn the relationship between the robot movement and the actual response of an object. The robot acquires this knowledge by randomly exploring the environment and without having any prior knowledge about the action. Initially, the robot was only able to move along straight lines in random directions. Action knowledge is stored in a neural network that encodes the effects of the exploratory movements with respect to the shape properties of the object (in a concrete example, point of contact on the object boundary and the angle of poke). Such self emergent processes are essential for the early cognition. The proposed process has been implemented and tested on the humanoid robot Hoap-3.

The estimated transformation functions are currently object-specific. To generalize the acquired action

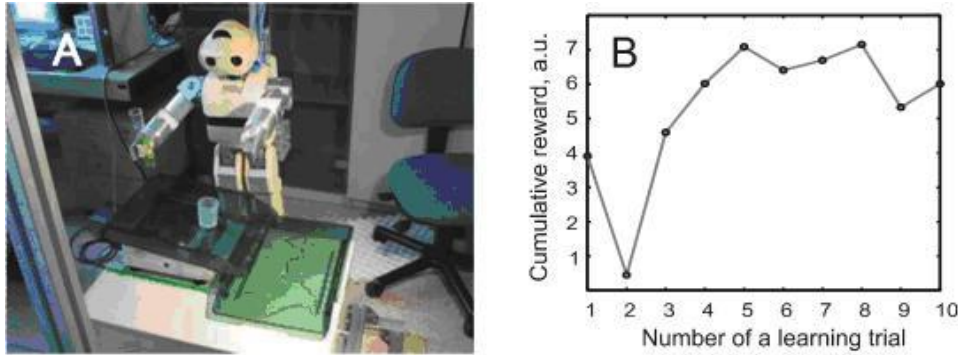


Figure 9: Robot pouring experiment. Setup of the experiment (A) and cumulative reward (in arbitrary units, a.u.) obtained over 10 learning trials. (B)

knowledge to a number of objects, we are developing a more general neural network which takes not only object location but also shape parameters as input. For example, this can be achieved by using the binarized object image as input to the network (instead of just point and angle of contact that were used in the poking example). In this way the system can acquire action knowledge that does not need to be learned separately for all objects.

3.4 Filling

Let us consider a glass filling task. We have a robot with a container full of liquid in the gripper, and want the robot to fill a glass standing on a table. Filling a glass is a good example of a task well suited for reinforcement learning (RL) as one does not know in advance how to position with high enough accuracy the robot's wrist in respect to the glass. This is due to the complicated physical process of liquid running out of a container and the inaccuracy of humanoid robot arms (industrial arms are here not considered, where one could indeed make an accurate enough model!). Thus, the correct position of the wrist can only be easily attained by learning. For RL the reward is the amount of liquid getting into the glass. For RL it is advantageous when not only full rewards (all liquid poured gets into a glass), but also partial reward (part of the liquid poured gets into a glass) are available. Here we have exactly that situation where in a non-optimal pouring position part of the liquid will get into the glass, while the rest will be spilled. Consequently, through reinforcement learning we can obtain a correct glass filling movement of a robot. Initial pre-positioning for pouring can be obtained through visual servoing (wrist close to the glass), or through demonstration.

Experiments were performed with the HOAP robot (cooperation between BCCN and JSI). We were operating with 3D coordinates of the robot wrist (task space), and on top implementing a standardized wrist movement for pouring. In the first set of experiments we were obtaining knowledge about the degree of filling of the glass through weighing, but later on the task is to be transferred to using visual analysis instead. For the setup of the experiment see Fig. 9 A. In Fig. 9 B we show the cumulative reward obtained in our learning experiments over ten trials. One trial includes moving over a learned trajectory (starting in the first trial with a random trajectory) with 7 to 10 attempts to pour on the way. One can see that near optimal performance is obtained in just 4-5 trials. This is a realistic scenario for a robotic application, where one would not be allowing hundreds of trials for learning. Future work is directed towards more fluent arm movements and generalization of learning towards more variable pouring situations.

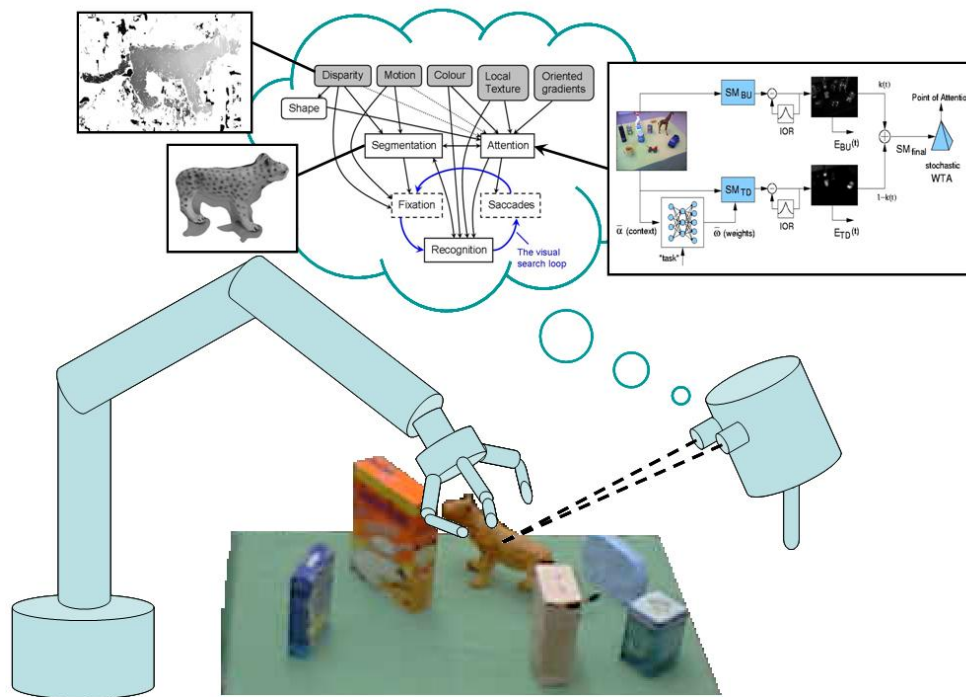


Figure 10: Illustration of an attention system described in [L].

3.5 Multi Cue–Affordance Relations

In this project, a robot vision system (see figure 10) needs to be able to autonomously acquire and suitably represent the environment in which it is operating. Thus, such a system needs the abilities to divide the world into *things*, create representations of observed *things* for later association and manipulation, and continuously update these representation as new data becomes available. A representation can either be short-lived and survive only a short sequence of actions, or permanent, if interactions with a 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.

As pointed out in the attached reference [L], the amount of perceptual data arriving through a visual system easily becomes overwhelming. 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 are 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. We call such a mechanism visual attention. Unfortunately, relevancy is not a static measure, but depends on the context, on the scene in which the robot acts and on the tasks the robot is performing. Consequently, there is a need for the attentional system to adapt to context changes. A 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 its current execution. Our OAC approach suggests an attentional mechanism that systematically relates perception and action, and it is indeed possible to derive attentional object-selection mechanism from action-planning processes (see Figure 11).

This motivates the consideration of object affordances not only from the interactive point of view (like grasping, pushing, filling), but also from the perceptual perspective (like fixating, processing) that factually precedes any manipulative action. In [L], it is presented how multiple visual cues and dense stereo can be used in a complementary way for detecting and attending objects in a general scene. The system is further used to extract object attributes such as those related to its shape. Detecting and storing object

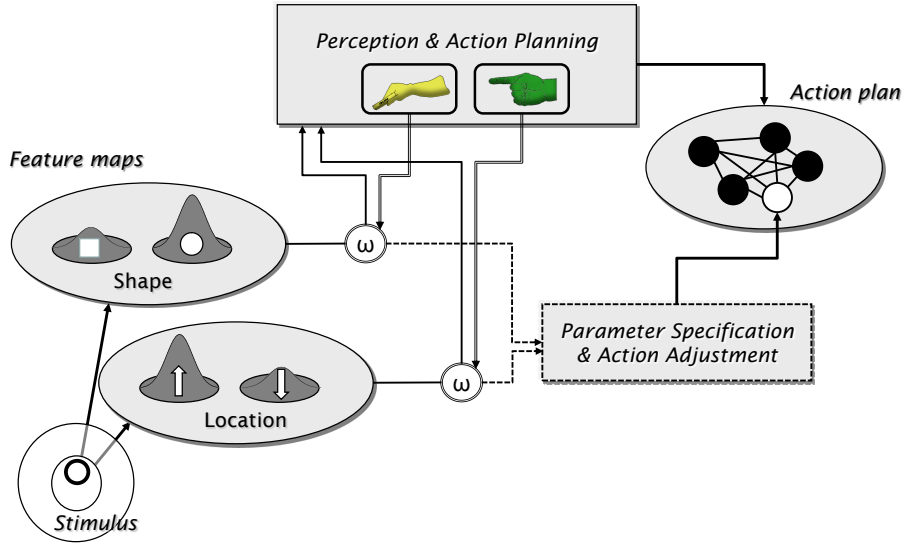


Figure 11: A process model of action-induced attention. Stimulus information coded on feature maps is directly fed into low-level action control (specifying the parameters not provided by high-level planning) but weighted according to the relevance of the given feature dimension for the currently planned action; i.e., planning a pointing action increases the weight of location information while planning a grasping action increases the weight of shape information [G].

attributes not only brings us from things to objects in a general OAC definition. In addition, it allows for further implementation of long and short term memory related to objects and actions being applied to them. The referenced system architecture describes connections and benefits of two distinctive grasp generation techniques in close relation to other modules, e.g. an attention system, a reasoning system, or the embodiment of the robot [H].

3.6 Rule System - Actions Affordance

The Rule System permits to generalize over objects to afford actions execution by learning the relevant attributes necessary to apply an action successfully [A, B]. Rules code at every moment how probable an action could be afforded with the experience acquired so far. The set of rules applicable to an object o constitutes an *OAC* where each rule selected for a concrete object o_i is an instantiation *iOAC*. Thus, given an object instantiation o_i , the system is able to predict how probable o_i afford the execution of an action.

The experience obtained from the instantiation is used to update the probabilities of the action affordance (P_{rule}^+) or failure (P_{rule}^-),

$$P_{rule}^+ = \frac{1}{2} \left(1 + \frac{n_{rule}^+}{n_{rule}^{total}} - \frac{n_{rule}^-}{n_{rule}^{total}} \right) \quad (1)$$

$$P_{rule}^- = \frac{1}{2} \left(1 + \frac{n_{rule}^-}{n_{rule}^{total}} - \frac{n_{rule}^+}{n_{rule}^{total}} \right) \quad (2)$$

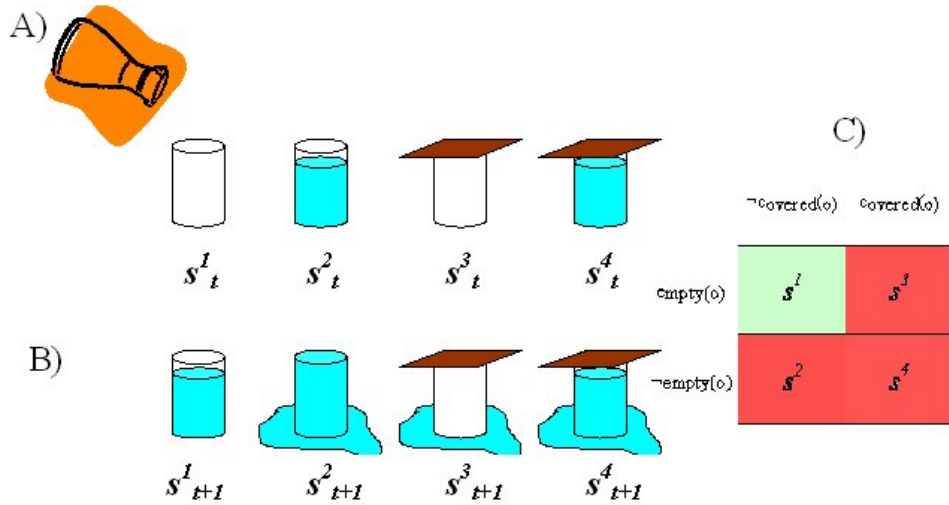


Figure 12: A) Initial states. B) States after action *fill* execution. C) State space of initial states where action *fill* is afforded (painted in green) or failed (painted in red).

where n_{rule}^{total} is the total number of all the possible states in which the rule could be involved.

With these formulas a high P_{rule}^+ is a confident indicator of a good chance of obtaining the prediction because the probabilities are based on densities of samples and not on relative frequencies, and assign to unexplored states the same chance to result in a successful or a failure. Therefore, a *rule* fed only with a few successful experiences has a probability of a success only a little higher than 0,5.

Rule set is progressively refined from experience using a general to specific constructive learning, and a memory based approach [B]. Whenever a rule has high uncertainty in its prediction (prob. close to 0,5) and large confidence (high density of samples), it is refined by generating new specializations of the rule using the information gain criterion.

Figure 12 illustrates the method with a very simple example using the affordance of an action *fill* over an object *glass*. The action *fill* is referred in this case to simply pouring a fix amount of liquid on the specific position of the glass. The attributes considered to evaluate the affordances are the boolean attributes $A_t = \{\text{empty}(\text{glass}), \text{covered}(\text{glass})\}$ for the initial attributes set, and $A_{t+1} = \{\text{empty}(\text{glass}), \text{clean}(\text{table})\}$ for the outcome evaluation. The action *fill* is afforded when $A_{t+1} = \{\neg\text{empty}(\text{glass}), \text{clean}(\text{table})\}$. Table 2 presents three examples of rules fed from the experience of situations s^1 and s^2 . As seen from the table, $rule_k$ is the one that most likely will afford the action *fill* in future experiences of situation s^1 .

rule	A_t	Ini states exp	n^+	n^-	P^+	P^-
<i>i</i>	$\text{empty}(\text{glass})$	s^1	1	0	0,75	0,25
<i>j</i>	$\neg\text{covered}(\text{glass})$	s^1, s^2	1	1	0,5	0,5
<i>k</i>	$\text{empty}(\text{glass}), \neg\text{covered}(\text{glass})$	s^1	1	0	1	0

Table 2: Examples of rules for action *fill* affordance ($A_{t+1} = \{\neg\text{empty}(\text{glass}), \text{clean}(\text{table})\}$).

We can conclude from this section that the outlined system is suitable for incremental approaches as the probabilities are based on densities of samples which avoid biased estimations with few experiences. It is also appropriate for real time performance since the simplicity of the updating formulas permits to rapidly have good estimations about the affordance of an action. It is important to mention that rules generated not only permit to evaluate single action affordance but also sequences of actions affordance when they are applied on the same object (see [A, B]). Finally, the last issue to remark is that the system developed so

far only deals with discrete representation of actions and objects. Further extension would incorporate also continuous attributes and actions.

4. Conclusion

In this delivery, we have presented work performed in WP4 in the last year. We have described the underlying representations on which generalization is taking place as well as the learning of concrete object–action associations on different level of the processing hierarchy which will become integrated in one system for the final review. A particular focus of the final year will be the learning of the association of object parts to actions as a tool for generalization across objects.

Attached Papers

- [A] A. Agostini, E. Celaya, C. Torras, and F. Wörgötter. Action Rule Induction from Cause-Effect Pairs Learned Through Robot-Teacher Interaction. In *In Proc. of the International Conference on Cognitive Systems, CogSys 2008. (Karlsruhe, Germany)*, pages 213–218, 2008.
 - [B] A. Agostini, F. Wörgötter, E. Celaya, and C. Torras. On-Line Learning of Macro Planning Operators using Probabilistic Estimations of Cause-Effects. Technical report, IRI-TR 05/2008. Institut de Robòtica i Informàtica Industrial. UPC-CSIC. (Barcelona, Spain), 2008.
 - [C] Emre Başeski, Dirk Kraft, and Norbert Krüger. A hierarchical 3d circle detection algorithm applied in a grasping scenario. *VISAPP*, 2009.
 - [D] B. Dellen, E.E. Aksoy, and F. Woergoetter. Segment tracking via a spatiotemporal linking process including feedback stabilization in an n-d lattice model. *Submitted to Pattern Recognition*.
 - [E] R. Detry, M. Popovic, Younes P. Touati, Emre Baseski, N. Krüger, and J. Piater. Autonomous learning of object-specific grasp affordance densities. *submitted to ICRA Workshop on Approaches to Sensorimotor Learning on Humanoid Robots*, 2009.
 - [F] Renaud Detry, Nicolas Pugeault, and Justus Piater. Probabilistic pose recovery using learned hierarchical object models. *International Cognitive Vision Workshop (Workshop at the 6th International Conference on Vision Systems)*, 2008.
 - [G] B. Hommel. Grounding attention in action control: The intentional control of selection. In *B.J. Bruya (ed.), Effortless attention: A new perspective in the cognitive science of attention and action.*, in press.
 - [H] B. Hommel and L.S. Colzato. When an object is more than a binding of its features: Evidence for two mechanisms of visual feature integration. *VISUAL COGNITION*, 17:120–140, 2009.
 - [I] K. Huebner and D. Kragic. Selection of Robot Pre-Grasps using Box-Based Shape Approximation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1765–1770, 2008.
 - [J] D. Omrčen, A. Ude, , and A. Kos. Learning primitive actions through object exploration. *Int. Conf. on Humanoid Robots, Daejeon, Korea*, pages 306–311, 2008.
 - [K] Mila Popovic, Dirk Kraft, Leon Bodenhagen, Emre Baseski, Nicolas Pugeault, Danica Kragic, and Norbert Krüger. A strategy for grasping unknown objects based on co-planarity and colour information. submitted.
-

- [L] 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*. Submitted 2008.
- [M] K. Welke, T. Asfour, and R. Dillmann. Active multi-view object search on a humanoid head. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009)*, 2009.

References

- [1] R. Detry and J. Piater. A probabilistic framework for 3d visual object representation. *IEEE PAMI*, submitted.
- [2] C.v. Ferber and F. Wörgötter. Clustering and recognition. *Phys. Rev.*, E 62:1461–1464, 2000.
- [3] Richard Alan Peters II, Kimberly E. Hambuchen, Kazuhiko Kawamura, and D. Mitchell Wilkes. The sensory ego-sphere as a short-term memory for humanoids. In *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2001.
- [4] K. Welke, T. Asfour, and R. Dillmann. Object separation using active methods and multi-view representations. In *Proc. IEEE International Conference on Robotics and Automation ICRA 2008*, pages 949–955, 19–23 May 2008.
- [5] K. Welke, E. Oztop, G. Cheng, and R. Dillmann. Exploiting similarities for robot perception. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 3237–3242, Oct. 29 2007–Nov. 2 2007.
- [6] Matthew Wright, James Chodzko, and Danny Luk. *Biologically Motivated Computer Vision*, chapter Development of a Biologically Inspired Real-Time Visual Attention System, pages 779–785. Springer Berlin / Heidelberg, 2000.
-

Action Rule Induction from Cause-Effect Pairs Learned through Robot-Teacher Interaction

Agostini A., Celaya E., Torras C. and Wörgötter F.

Abstract— In this work we propose a decision-making system that efficiently learns behaviors in the form of rules using natural human instructions about cause-effect relations in currently observed situations, avoiding complicated instructions and explanations of long-run action sequences and complete world dynamics. The learned rules are represented in a way suitable to both reactive and deliberative approaches, which are thus smoothly integrated. Simple and repetitive tasks are resolved reactively, while complex tasks would be faced in a more deliberative manner using a planner module. Human interaction is only required if the system fails to obtain the expected results when applying a rule, or fails to resolve the task with the knowledge acquired so far.

I. INTRODUCTION

IN this work we are facing the problem of decision making for a multitask robot embedded in a human environment that should rapidly learn to perform tasks by interacting with humans, in an on-line way, and without any previous knowledge of the world dynamics or the tasks to be performed.

From a very general point of view, we must consider two alternative approaches to the goal of building an intelligent agent: the deliberative and the reactive approaches. The deliberative approach began with the very birth of AI, and it is based on the principle of rationality [1], which states that "If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.". The proponents of the knowledge-based systems using the principle of rationality soon realized that there are a number of important shortcomings with this approach, ranging from the frame problem [2], the difficulty of building a large enough database of knowledge providing the grounds for common sense, and the theorems stating the complexity of planning for even some of the simplest kinds of logical problems.

Later, also the symbol grounding and related problems [3] entered the scene. As a response to this, the proponents of

the new AI [4] advocated for the reactive approach, in which the knowledge level was completely absent. In this approach actions are not driven by the rationality principle, but triggered by the current situation, and not guided by any specific purpose, but simply as a set of instincts carefully organized to accomplish a specific task.

While reactive approaches have proved to be valid for many low-level tasks, we think that the kind of intelligent behavior we expect from a service robot, like a kitchen assistant, cannot be the result of purely reactive processes. We want the robot to promptly accomplish the task required by the user, and this means that its actions must be goal-driven, and not just situation-driven. We expect the robot to be able to produce new behavior in response to a new goal using its knowledge of the situation and the effects of its actions, but it is clear that a reactive system will only be able to act according to already acquired behaviors.

A number of hybrid approaches have been proposed along these lines. Some of them propose a decision-making system that permits fast agent responses to new situations using reactive layers while the deliberative layers generate behaviors used later by the reactive modules [5]. Others let the low-level action control to be driven by reactive behaviors, which are selected or modulated by a higher deliberative layer [6], [7]. Finally, some works focus mainly on the generation of behaviors such as macro-actions [8], primitive behaviors [9], or activation rules [10], which store sequences of actions frequently used or difficult to calculate, to use them later as macro planning operators in a deliberative system.

In any of the previous cases a large amount of computation is usually required due to the need of exploring different acting behaviors to select one suitable for the task. The problem turns to be more complicated if the robot has no previous knowledge of the world dynamics and should perform learning while predicting what would occur with different behaviors. Incomplete knowledge has been tackled using techniques like incomplete planning [11], learning planning operators [12], [13], [14] or policy learning [15], but the drawback of computational complexity derived of the application of AI techniques is still not surmounted.

The aim of this work is to develop an integrated system in which reactive and deliberative components are both present, though not strictly separated, but smoothly combined, and where the world dynamics and behaviors are rapidly learned from scratch through a natural human-robot interaction.

This work is funded by the EU PACO-PLUS project FP6-2004-IST-4-27657.

A. Agostini is with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain (corresponding author) (phone: +34-93-401-5786; fax: +34-93-401-5750; e-mail: agostini@iri.upc.edu).

E. Celaya, is with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain (e-mail: celaya@iri.upc.edu).

C. Torras is with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain (e-mail: torras@iri.upc.edu).

F. Wörgötter is with the Bernstein Center for Computational Neuroscience, Göttingen, D37073, Germany (e-mail: worgott@bccn-goettingen.de).

As we want the agent to learn only the dynamics of the world relevant for its purpose, the world exploration is guided by a teacher. It is very simple for humans to know which action to perform in a situation given a plain task, like a kitchen task, but it could be much more complicated to explain a priori all the sequences of actions that should take place in all the possible situations. It might also be difficult for a human to detail all the conditions that should be taken into account to afford a desired cause-effect for all the possible situations. In this work we take benefit of the human capabilities of explaining cause-effect relations in currently observed situations to efficiently generate knowledge for decision making in a multitask robot. The idea is based on Piaget's theory of cognitive development which claims that children gradually acquire knowledge of cause-effect relations by repeatedly executing processes and sequencing actions to reach goals.

This work is organized as follows. Section II explains the outline and main elements of the method proposed. Section III presents a demo application and clarifies some concepts explained in Section II. In Section IV the algorithm is delineated in pseudo-code. A brief discussion of the ideas and concepts of this work in the context of the European project PACO+ [16] is developed in Section V. Finally, section VI delineates some conclusions and future works.

II. OUTLINE OF THE METHOD

In this work a decision making system is proposed where the action behaviors are generated using simple cause-effect relations learned with the help of a teacher. The learned behaviors are used either reactively or deliberately depending on the complexity of the task requested.

We will define a behavior (or rule) as a set of preconditions, a sequence of actions, and the final expected outcome. The preconditions are a set of necessary conditions or perceptions that must be observed before the rule can be applied, and the expected outcome is a series of effects that will be obtained after the execution of the rule. The action sequence may consist of a single elementary action in the simplest rules (the cause-effect relation for that action) or a list of actions, each one expressed in turn as a cause-effect.

A general overview of the proposed method is the following. Given a goal, the agent tries to apply any of the existing rules in a reactive way to reach it from the current situation without any deliberation. If more than one rule is retrieved, the one with fewer actions in its sequence is applied. If a reactive behavior is not possible, then the agent tries to generate a plan using the existing rules as planning operators.

If both the reactive and deliberative modules fail to return a behavior, as a consequence of an incomplete knowledge, the agent asks the teacher about which action or actions to perform. The agent executes every instructed action and generates a first approximation of the involved cause-effects by observing the changes in the environment. Then the agent

generates a rule with the sequence of the generated cause-effects.

On the contrary, in the case that the agent is able to find a behavior with the reactive or deliberative module, then it executes and evaluates it at the level of each cause-effect in the related sequence. If any of the outcomes obtained is different from the one expected, the agent will ask the teacher for explanations about which conditions prevented the correct outcome of the cause-effect to occur. With the teacher explanation the agent automatically corrects the cause-effect structure as well as all the rules that apply this cause-effect in their sequences performing a large updating of the knowledge base with a little teacher interaction.

A. Notation

We assume that the agent has a set of N sensors that measure some features of the environment. The value of sensor i is called an observation o_i . A world state SO is formed by the set of observations o_i , $SO = \{o_1, o_2, \dots, o_N\}$.

Each of these sensors is internally represented by the agent as a detector d_i that could take different discrete values d_{ij} , called conditions, depending on the sensed value o_i . An internal agent state S is constituted by a set of conditions d_{ij} , $S = \{d_{1j}, d_{2k}, \dots, d_{Ni}\}$.

At every moment the agent is able to perform any of the k actions from the set $A = \{a_1, a_2, \dots, a_k\}$.

The function that maps the sensor observations to conditions is called the *perception function* (PF). As we will explain later the PF could be updated while the learning process is running, permitting the management of the uncertainties, inherent to real environments.

The most elementary rule consists of a cause-effect relation and reflects how a change is obtained using a single action and what preconditions are necessary to afford that change. We formally represent a cause-effect cec_i using a tuple that consists in a subset P_i of state conditions called the preconditions of the cec_i , an action a_i from the set of actions A , and a subset O_i of state conditions denoted as the expected outcome of the cec_i .

$$cec_i = \langle P_i = \{d_{g_j}, \dots, d_{m_l}\}, a_i, O_i = \{d_{k_l}, \dots, d_{p_q}\} \rangle \quad (1)$$

In the same way, a rule R_j is described using a tuple that consists of a subset P_j of state conditions called the preconditions of the rule R_j , a sequence of cec 's $CECS = (cec_k, cec_i, \dots, cec_m)$, and a subset O_j of state conditions denoted as the expected outcome of the rule.

$$R_j = \langle P_j = \{d_{i_h}, \dots, d_{m_l}\}, CECS, O_j = \{d_{k_l}, \dots, d_{p_q}\} \rangle \quad (2)$$

In our approach, the expected outcome serves two purposes: it will be used by a goal-achieving deliberative system for planning, and by a learning system to improve rule descriptions. Every time the expected outcome is different from the observed we will say that the robot gets a *surprise*.

B. Learning Rules

When the knowledge base of the system doesn't permit to find a rule, or a sequence of rules, to be applied in an experienced situation, the teacher instructs the robot about which action or sequence of actions to execute. Then, the robot executes every instructed action generating a first approximation of the involved cause-effects, and afterwards builds a rule using the sequence of the generated *cec*'s.

1) Generating *cecs*

The robot generates a first approximation of the cause-effect observing the conditions that change in the states before and after the execution of the instructed action a . If we call the state before the action execution S^{prior} and the state after the action execution S^{post} the new cec_{new} is:

$$cec_{new} = \langle P_{new}, a, O_{new} \rangle \quad (3)$$

where,

$$P_{new} = \{ d_{ij} \in S^{prior} \mid d_{ij} \notin S^{post} \} \quad (4)$$

$$O_{new} = \{ d_{kl} \in S^{post} \mid d_{kl} \notin S^{prior} \} \quad (5)$$

The preconditions of the cec_{new} so formed could be incomplete in the sense that there could be conditions that do not change before and after action execution, but are also necessary to produce the changes observed (for example, the density of an object that prevents its deformation, the friction of a surface that prevents an object displacement, etc.). In these cases the teacher would explain which conditions are missing to obtain the expected outcome.

2) Generating Rules

The generated *cec*'s are used to create rules that will contain the *cec*'s sequence. The preconditions $P_{R_{new}}$ of the rule R_{new} formed should ensure the occurrence of the *cec*'s preconditions in the proper order. For those detectors that take only one condition value during the sequence, it is straightforward that this value should also appear in the rule preconditions. If there is more than one condition for a particular detector, the one closer to the origin of the sequence should occur first, and this one should be in the rule precondition. Therefore, the rule preconditions can be obtained directly by back-propagating with replacement all the preconditions of the *cec*'s departing from the last *cec* to the first one. In contrast, to deduce the final outcome of the *cec*'s sequence, and hence of the rule $O_{R_{new}}$, we should take into account the last changes produced in each detector. Therefore, if there is more than one condition for a detector, the one that should be considered for $O_{R_{new}}$ is the farthest from the origin. We can obtain all the conditions of the rule outcome again by back-propagating the conditions of the *cec*'s outcomes, but now without replacement departing from the last *cec* to the first one. The process of rule generation is illustrated in Section III.

There are two remarkable aspects. The first one is that, assuming all the proper preconditions are considered, the

rules would produce the expected outcome in all the situations where the respective preconditions are present, despite some of these situations not having ever been experienced before. Therefore, rules perform generalization over all the situations where the corresponding sequence would take place. The second notable characteristic is that, for a given sequence CECS of *cecs*, as many rules as sub-sequences in CECS could be generated, using the initial and final *cec* of the sub-sequences as the initial and final points for the back-propagation procedure. The subset of rules actually generated depends on the criterion adopted. For instance, the robot could be required to only learn how to reach the goals specified by the teacher, leading to the generation of only the subset of rules consisting of every sub-sequence from an intermediate situation to the goal.

C. Rule Correction

During the execution of a rule the robot can get a surprise if one of the involved *cecs* results in an unexpected outcome. Then, the teacher "explains" which preconditions prevented the expected outcome to occur. The reason of the surprise could be produced by either a missing condition in the precondition part or by a wrongly interpreted condition due to a problem in the perception function PF . In both cases the teacher tells the robot which conditions are responsible for the failure, specifying the detectors and the corresponding values. The explanation given is used to update the PF and to correct the *cec*. The explanation could be indeed incomplete, not specifying all the conditions that would prevent the expected outcome to occur, but only those that the teacher is capable of identifying at that moment as the ones responsible for the surprise. This is accepted as far as the teacher is able to realize in future observations the other conditions that are responsible for the failure.

After the *cec* correction, the rules correction is simple and straightforward. It is performed by updating all the rules that contain the corrected *cec* in their sequences just by back-propagating the explained conditions as explained in the rule generation section.

D. On Learning to Perceive

We want to briefly remark the underlying idea about how the perception function PF could be updated using the teacher explanations. The idea is expressed in the scope of simple applications (like the one presented in Section III) where the perceptions of the robot could be derived by the sensor observations using a probabilistic approach.

If we assume that the sensor observations o_i are continuous variables with uncertainties and non-stationarities, the way to correctly map a value o_i to a condition d_{ij} is difficult to establish a priori. It is possible to face this matter through a probabilistic approach that for each condition d_{ij} permits to infer how probable it is that a sensed value o_i is interpreted as d_{ij} . Then, for a particular observed value o_i , the condition d_{ij} perceived is the one with highest probability in o_i . The estimated statistic values

related to a condition d_{ij} could be updated using the teacher explanations on this condition using the corresponding observed value o_i . This updating permits the teacher also to explain the robot how to interpret the world.

III. DEMO APPLICATION

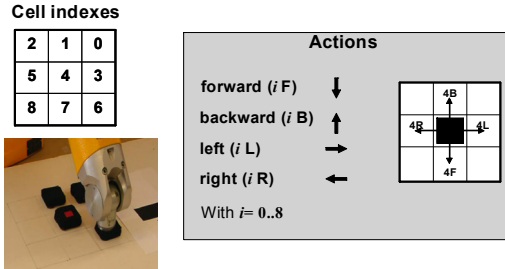


Fig. 1. Demo application elements.

Figure 1 shows a schema of a simple real world application implemented in a Staubli arm that permits to visualize the important aspects of the method performance. The application consists in an environment with 9 cells configured in a 3 by 3 grid world. Each cell could contain a black box or be empty. The amount of boxes that can be placed in the grid ranges from 1 to 8. Among all the boxes there is one target box marked with a red label. The task consists in placing the target box into a goal cell without taking any box outside of the grid. To this purpose the arm can move, when it is possible, any box from its current cell to one of the contiguous cells in straight line (diagonal movements are not allowed). Movements that take any box out of the grid cannot be performed.

A cell is considered as a detector in the state representation. The state is represented graphically in the examples, where a black cell represents a box in that cell, a white represents an empty cell and a black with a red mark

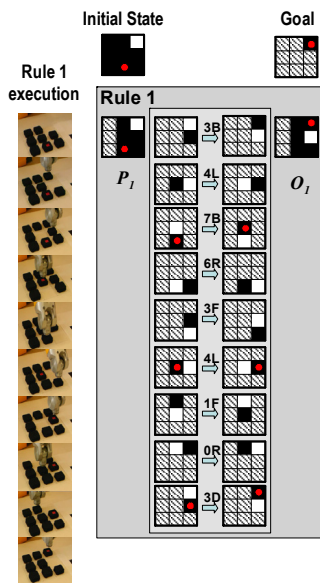


Fig. 2. The largest rule generated after the instructions *iseq1* to take the target box from cell 7 to cell 0.

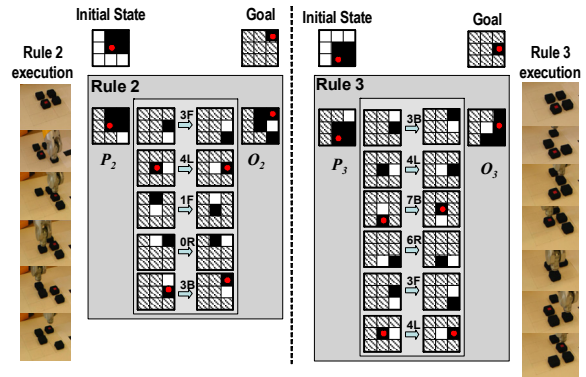


Fig. 3. Execution of two rules generated from *iseq1* under different goals requirements.

represents the cell containing the target box. Dashes cells mean “don’t care” if there is either a box or an empty space. Figure 1 explains the possible actions and how the cells are indexed to reference detectors and actions.

Before presenting some results we would like to mention that the rule generation criterion adopted for this example is to generate as many rules as sub-sequences there are in the instructed sequences. The robot started the experiments without any previous knowledge. Due to space restrictions the process of instructions is not shown graphically but mentioned during the descriptions of the experiments.

The first instruction received by the robot was to move the target box from cell 7 to cell 0 with the grid full of boxes except cell 0 which was empty. This instructed sequence will be referenced in the following as *iseq1*. Figure 2 shows the largest rule generated from *iseq1* and snapshots of the rule execution given an initial state and goal where the rule was applicable. Figure 3 shows two more rules generated with *iseq1* executed under different requirements of goals and with initial states never experienced before by the robot. The possibility of resolving situations never experienced elucidates the generalization capabilities of the method.

We now instruct the robot to move the target box from cell 5 to cell 4, when cell 4 is occupied and cell 7 is empty.

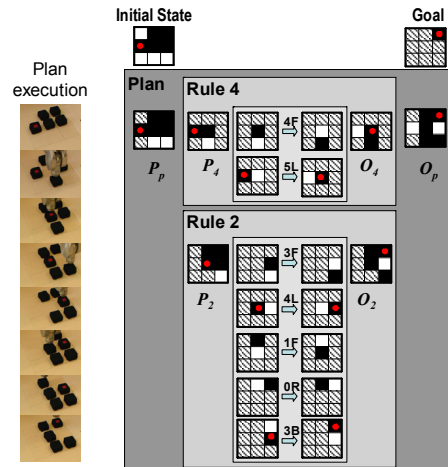


Fig. 4. Plan that linked rule 4 and rule 2 to fulfill a given requirements of initial state and goal.

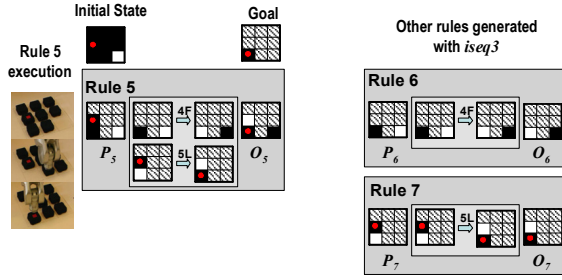


Fig. 5. Rules generated with *iseq3*.

The instructed sequence is denoted as *iseq2*. As a consequence of *iseq2* the robot generated rule 4 (see figure 4). Next, to show how the planner module is activated when no rule is reactively triggered the target cell was placed again in cell 5 but now four more boxes were added in the grid configuring an initial situation shown in figure 4. The robot was then asked to take the target cell from cell 5 to cell 0. For these requirements there was no rule in its database that permitted a reactive behaviour. The planner module was then activated and a plan, that linked rule 4 with rule 2, was found and executed. Figure 4 also suggests how a plan could be transformed into a new rule using the condition propagation.

A. Surprise and Explanation

In this section we exemplify how a surprise arises and how the explanations are used to correct the incomplete *cec* and the rules that involve it. First we instructed the robot to move the target box from cell 5 to cell 8 when cells 8 and 7 are occupied and cell 6 is free. This instruction is referred to as *iseq3*. Figure 5 shows the execution of the largest rule generated with *iseq3*, as well as the other two rules generated with the same sequence. Note that for the first action instructed the robot pushed boxes in the cells 8 and 7 but the state representation for cell 7 remained the same before and after the action execution. Hence the generated *cec*, which extracted only the conditions that changed, initially contained a “don’t care” in that position.

Afterward, in figure 6, we made the robot to face a

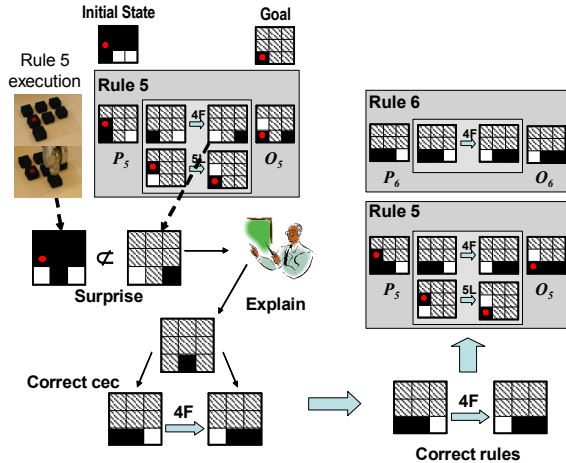


Fig. 6. A schema of the surprise-explain-correct process.

problem where the initial state and goal triggered rule 5. The first *cec* execution led to a surprise as the obtained outcome was not included in the expected ones. The teacher then explained that a black in cell 7 should also be considered and the robot corrected the *cec* as well as the involved rules.

Finally, we made the robot face the same problem that previously resulted in a surprise but then no rule could be applied reactively. Nevertheless, the robot found a plan using one of the rules generated with *iseq1* (rule 8) and the recently corrected rule (rule 5) as illustrated in figure 7. The plan found is not the optimal way to solve the problem because the robot was only able to use the limited knowledge acquired so far. It is important to mention that, in case many plans are found, the robot uses the same criterion as with the rules, i.e., it selects the one with fewer actions.

IV. SKETCH OF THE ALGORITHM

In this section we present the whole method in pseudo-code. It is important to remark that, in this first approach, we let the teacher control the rule generation by the instruction given. The teacher will instruct a single action when no sequence is convenient to be merged in a rule, and will instruct a sequence of actions for repetitive sequences.

A. Pseudo-code

```

INIT system RR={}, LCECS={}, CECS={}
Define GOAL
Sprior=PF(SOprior)
WHILE goal is not reached
  RR: rules that connect Sprior to GOAL
  IF RR is not empty (Reactive)
    Select rule of RR with fewer cecs in CECS
    Execute CECS
  ELSE (RR is empty)
    Try to find a PLAN with the rules.
    IF PLAN is possible (Deliberative)
      Execute CECS
    ELSE (If plan is not possible)
      Teacher instructs actions
      FOR each action instructed,
        Sprior = PF(SOprior)
        Execute action
        Spost = PF(SOpost)
        GENERATE new cec using Sprior and Spost
        APPEND the cec to LCECS
      END FOR
      GENERATE RULES using LCECS
    END ELSE (planning not possible)
  END ELSE (RR is empty)
  Sprior=PF(SOprior)
  Teacher supervises if Sprior is well perceived
  IF Sprior is wrongly perceived
    Teacher explains bad conditions
    UPDATE PF
    Correct Sprior
  END IF
END WHILE (goal is not reached)

```

1) Execute CECS

```

FOR each cec in CECS
  Execute action
  Spost=PERCEIVE(SOpost)
  IF not the expected outcome (SURPRISE)
    Teacher explains bad/missing conditions
    Correct cec with teacher explanations
    Correct rules containing the cec
    Update PF
  EXIT FOR
END FOR

```

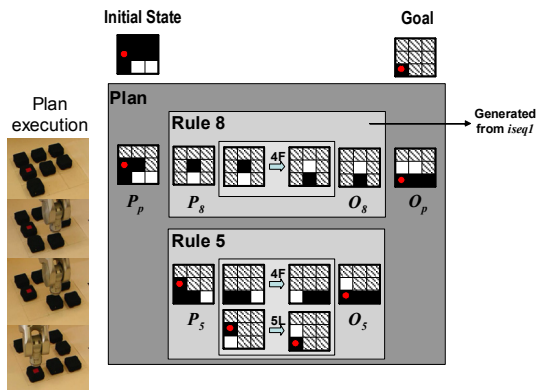


Fig. 7. Plan execution involving one rule generated with *iseq1* and the corrected rule generated with *iseq3*.

V. DISCUSSION IN THE CONTEXT OF PACO+

Most of the “learning to act” approaches are based on human learning and cognition capabilities. Despite these approaches present many differences among them, they all establish a direct relation between perceptions of the agent, coded mainly as states, and actions. In contrast to the amount of approaches developed, only few attempts were aimed at creating a common framework that permits to consistently relate the learning to act approaches with the human cognition capabilities for learning and acting. One of these attempts is the concept of object-action complexes (OACs) [17] that has been evaluated and developed by the European PACO+ consortium [16]. Briefly, the OAC concept claims that the world contains undistinguished “things” meaningless for the agent that only become meaningful “objects” through actions and tasks, where the objects are described by the properties relevant for the fulfillment of the final desired outcome through the action.

We believe that the explicit coding of the world conditions and actions through rules and cause-effects presented above is suitable for a first insight in the study and refinement of the OAC concept. One of the reasons is that the elements of these structures could be directly associated with the main elements of the OACs concept formulated so far. Another reason is that they permit a direct association with the human cognition capabilities through the explicit declaration of the abstract meaning of the conditions of the world, and hence a better understanding and a faster evaluation of the results.

VI. CONCLUSION AND FUTURE EXTENSIONS

Despite the advantages presented in using simple human instructions for learning to perform tasks, the system should be also able to perform a task without the help of any human in case there is none available. This could be fulfilled by giving the instructions and explanation by other embedded automatic systems. The instruction could be given by an incomplete planner establishing some criterion for rule generation with a measure of the frequency of usage and the

amount of computational process needed to generate a given plan. The explanation could be replaced by a constructive learning system where, for instance, a memory-based system could permit to infer which conditions are responsible for the surprise [12], [13], [14]. We believe that the presented method establishes a very suitable platform for future extension to develop a robust decision-making system for a complex robot interacting in a human environment.

REFERENCES

- [1] A. Newell. “The knowledge level”. *Artificial Intelligence*, 18(1), 87-127, 1982.
- [2] J. McCarthy and P.J. Hayes. “Some Philosophical Problems from the Standpoint of Artificial Intelligence”, in B. Meltzer and D. Michie eds., *Machine Intelligence*, Edinburgh: Edinburgh University Press, 1969, pp. 463-502.
- [3] S. Harnad. “The Symbol Grounding Problem”. *Physica D* 42: 335-346, 1990.
- [4] R. Brooks. “Intelligence without representation”. *Artificial Intelligence*, 47, pp. 139-159, 1991.
- [5] M. Lemaitre, G. Verfaillie. “Interaction between reactive and deliberative tasks for on-line decision-making” presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [6] E. Gat. “On three-layer architectures” in D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors. *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, pp. 195-210, 1998.
- [7] M. J. Schoppers. “Universal Plans for Reactive Robots in Unpredictable Environments”, in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI 87)*, Milan, Italy, 1987, pp. 1039-1046.
- [8] M. Newton, J. Levine. “Evolving Macro-Actions for Planning”, presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [9] M. Nicolescu, M. Mataric. “A Hierarchical Architecture for Behavior-Based Robots” in *Proc. of the 1st Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, Bolgna, Italy, 2002, pp. 227-233.
- [10] D. Rao, Z. Jiang, Y. Jiang. “Learning Activation Rules for Derived Predicates from Plan Examples” presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [11] S. Yoon, S. Kambhampati. “Towards Model-lite Planning: A Proposal For Learning & Planning with Incomplete Domain Models” presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [12] X. Wang. “Learning planning operators by observation and practice”, in *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, USA, 1994.
- [13] T. Oates and P. Cohen. “Learning planning operators with conditional and probabilistic effects”, in *Proceedings of the AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996, pp. 86-94.
- [14] S. Benson. “Inductive learning of reactive action models”, in *Proceedings of the 12th International Conference of Machine Learning*, 1995, pp. 47-54.
- [15] R. Sutton and A. Barto. “Reinforcement Learning. An Introduction”. MIT Press, 1998.
- [16] <http://www.paco-plus.org>
- [17] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krüger and F. Wörgötter. “Object Action Complexes as an Interface for Planning and Robot Control”, presented at *IEEE RAS Int Conf. Humanoid Robot*, Genova, Italy, 2006.

On-line Learning of Macro Planning Operators using Probabilistic Estimations of Cause-Effects

Agostini A., Wörgötter F., Celaya E., Torras C.

Abstract. In this work we propose an on-line learning method for learning action rules for planning. The system uses a probabilistic approach of a constructive induction method that combines a beam search with an example-based search over candidate rules to find those that more concisely describe the world dynamics. The approach permits a rapid integration of the knowledge acquired from experience. Exploration of the world dynamics is guided by the planner, and – if the planner fails because of incomplete knowledge – by a teacher through action instructions.

Introduction

In the last years service robot applications are widening with the improvements in computer science techniques and the development of new technologies. These applications range from simple chores, like vacuum cleaners, to complex tasks requiring complex cognitive capabilities, similar to those forming the human capacity of performing complex tasks in real environments.

In this work we face the problem of decision making for a multitasking service robot embedded in a human environment that should rapidly learn to perform tasks in an on-line way, and without any previous knowledge of the world dynamics or the tasks to be performed.

The selection of which paradigm to apply relies on the characteristic of the problem faced. In general, there are two alternative approaches used to build an intelligent agent: the deliberative and the reactive approaches. The deliberative approach is based on the principle of rationality [1], and involves planning techniques in which actions are executed in accordance to plans built after reasoning about possible sequences to reach the goal. In reactive approaches [2] actions are not longer driven by the rationality principle, but just executed from already coded behaviours that lead to the goal without deliberation.

Both paradigms have drawbacks and advantages. In planning approaches the amount of deliberation could be very large even for the simplest kind of problem, mainly when the environment has complex dynamics. Additionally, deliberative techniques require a model of the dynamics of world, which in many cases is extremely difficult to code. This is not usually the case for reactive techniques which learn dynamics of the world automatically. Nevertheless, in reactive approaches drawbacks are caused by the limitation of their applicability to single goal tasks and by the requirement of large experience to reach an acceptable convergence. We propose a decision making technique that combines both approaches with the aim of diminishing the drawbacks of either one of them using the advantages of the other.

In general, reactive approaches have proved to be valid for many low-level repetitive tasks, while deliberative approaches are more suitable for high-level tasks where specified goals are rarely repeated. Usually, it is observed that high-level tasks can be performed by a succession of simpler low-level tasks.

We develop a method to learn on-line action rules that permit to reactively perform low-level tasks when they are executed as planning operators of high-level plans. These rules could significantly relieve the amount of deliberation as they might merge repetitive sequences of actions, or plans found with large computational cost. One remarkable aspect of the method is that there is no need for codification of the world dynamics as they are learned automatically while acting.

The learning method generates action rules using a constructive induction approach that combines a beam search with an example-based search [3] over candidate action rules to find those that more concisely describe the world dynamics. The approach permits a rapid integration of the knowledge acquired from experience. Exploration of the world dynamics is guided by the planner, and – if the planner fails because of incomplete knowledge – by the teacher through action instructions.

It is very simple for humans to know which action to perform in a situation given a plain task. But it could be much more complicated to explain a priori all the sequences of actions that should take place in all the possible situations. We take benefit of the human capabilities of knowing which action to perform in currently observed situations to efficiently generate knowledge for decision making in a multitask robot.

The idea of learning cause-effects is based on Piaget's theory of cognitive development [4] which claims that children gradually acquire knowledge of cause-effect relations by repeatedly executing processes and sequencing actions to reach goals. As we will see, action rules are created using learned cause-effect relations observed from experienced situations after actions executions. Cause-effects are not only expressed as a unique set of conditions that afford changes, but also as multiple set of conditions that have different chances to afford a change.

As suggested in previous works [5], [6] the explicit coding of the world conditions and actions through rules and cause-effects presented above is one possible instantiation of the concept of object-action complexes (OACs) [7], which is considered as the main block for building cognitive systems for a complex service robot [8]. In a few words, the OAC concept claims that the world contains undistinguished “things” meaningless for the agent that only become meaningful “objects” through actions and tasks, where the objects are described by the properties relevant for the fulfilment of the final desired outcome through the action. We believe that the contribution of this work is another step toward a formalization of the OAC concept as the probabilistic approach of cause-effect could be seen as how likely a thing is an object, where an object description now is not restricted to a unique set of conditions but multiple set of conditions that have different chances of affording the object functionality.

The learning module is embedded in a more general decision making system containing a planning module that uses the learned action rules as planning operators. The global system is based on a previous study [5] where the teacher guides the exploration of actions and also explains the world dynamics at the level of currently experienced cause-effects. The current contribution is an extension of [5] where dynamics of the world are automatically learned while performing the required tasks.

Other approaches propose a decision-making system that permits fast agent responses to new situations using reactive layers while the deliberative layers generate behaviors used later by the reactive modules [9]. Some let the low-level action control to be driven by reactive behaviors, which are selected or modulated by a higher deliberative layer [10], [11]. Finally, others focus mainly on the generation of behaviors such as macro-actions [12], primitive behaviors [13], or activation rules [14], which store sequences of

actions frequently used or difficult to calculate, to use them later as macro planning operators in a deliberative system.

In any of the previous cases a large amount of computation is usually required due to the need of exploring different acting behaviors to select the one suitable for the task. The problem turns to be more complicated if the robot has no previous knowledge of the world dynamics and should perform learning while predicting what would occur with different behaviors. Incomplete knowledge has been tackled using techniques like incomplete planning [15], learning planning operators [16], [17], [18] or policy learning [19], but the drawback of computational complexity derived of the application of AI techniques is still not surmounted.

System structure

As mentioned before, the decision making system has two main modules: a learning module that provides action rules in the form of planning operators, and a planning module that uses the learned operators. The action rules learned have STRIPS like structure [20] suitable to be used by any planner that can deal with them.

A general overview of the method is the following. Given a goal, the agent tries to generate a plan using the existing rules. If the planner fails to return a behavior, as a consequence of an incomplete knowledge, the agent asks the teacher about which action or actions to perform. The agent executes every instructed action and generates what we denote as a *probabilistic cause-effect* used to estimate the probability of the occurrence of changes under different sets of conditions. A probabilistic cause-effect is the main structure for learning and generation of STRIPS like cause-effects for planning purposes.

When the agent is able to find a plan then it executes and evaluates it at the level of each cause-effect. During cause-effects execution relevant experiences are stored as example situations which are used to learn the conditions that afford desired changes. For instance, if any of the outcomes obtained after a cause-effect execution is different from the one expected, a fact referred to as *surprise*, the experienced situation is memorized as a negative example for obtaining the expected change.

Whenever a sequence of cause-effects is successfully executed it could be memorized into a *rule* to relieve the reasoning load of the planner. Rules are essentially similar to cause-effects but instead of an action they contain sequences of actions, each one in turn expressed as cause-effect.

Figure 1 illustrates a general schema of the system.

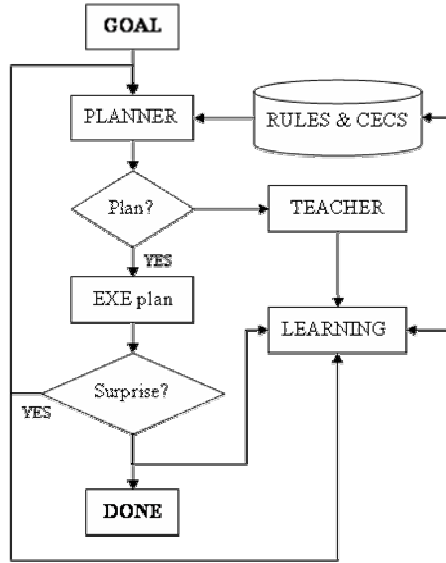


Figure 1. General architecture of the decision making system.

Notation

We assume that the agent has a set of N detectors d_i , $i=1..N$, that could take different discrete values d_{ij} , $j=1..|d_i|$, called conditions. A state s is constituted by a set of conditions d_{ij} , $s=\{d_{1j}, d_{2k}, \dots, d_{Nl}\}$. Any set of state conditions is denoted as a subspace ss .

At every moment the agent is able to perform any of the k actions from the set $A=\{a_1, a_2, \dots, a_k\}$.

We define a *probabilistic cec* ($pcec$) by a tuple containing a precondition part that involves two set of subspaces, the working (SS_w) and the candidate (SS_c) subspaces, and a list of example states L_s , an action part a_{pcec} , and the expected outcome O_{pcec} of the $pcec$, coded also as a subspace.

$$pcec = \langle \{SS_w, SS_c, L_s\}, a_{pcec}, O_{pcec} \rangle$$

Each subspace ss in SS_w and SS_c has associated three numbers,

- n_+^{ss} , counter for positives examples covered by ss .
- n_-^{ss} , counter for negatives examples covered by ss .
- n^{ss} , total number of possible states in the region covered by ss .

Where positive and negative examples are those states experienced, and stored under which the expected change O_{pcec} occurs after the execution of a_{pcec} or fails to be obtained, respectively. In a similar way, we refer to the probability of a positive example as the probability of obtaining the expected change, and the probability of a negative one for the converse.

Additionally, we formally represent a cause-effect cec_i using a tuple that consists in a subspace P_i called the preconditions, an action a_i , and a subspace O_i denoted as the expected outcome of the cec_i . The preconditions indicate under which conditions the cause-effect can be applied, and the expected outcome reflects the effects that will be obtained after its execution.

$$cec_i = \langle P_i = \{d_{gj}, \dots, d_{ml}\}, a_i, O_i = \{d_{kl}, \dots, d_{pq}\} \rangle$$

Thus, a cec can be seen as a formal instantiation of the more abstract OAC as discussed in the Introduction.

In the same way, a rule R_j is described using a tuple that consists in a subspace P_j called the preconditions of the rule R_j , a sequence of cec 's $CECS = (cec_k, cec_i, \dots, cec_m)$, and a subspace O_j denoted as the expected outcome of the rule,

$$R_j = \langle P_j = \{d_{ih}, \dots, d_{ml}\}, CECS, O_j = \{d_{kl}, \dots, d_{pq}\} \rangle$$

Rules can therefore be considered as chains of OACs. In our approach, the expected outcome serves two purposes: it will be used by a goal-achieving deliberative system for planning and to evaluate the outcomes.

Generating $cecs$ from $pcecs$

It is not possible to use the probabilities estimations directly for planning because the system is not designed for a probabilistic planner but for a deterministic one. Thus, for plan generation only $cecs$ and rules can be used. In this section it is explained how $cecs$ are obtained from $pcecs$.

To generate $cecs$ from $pcecs$ we only use subspaces of the set of working subspaces SS_w . First, an estimation of the probabilities for a positive and a negative example for each subspace in SS_w need to be obtained. There are many ways of estimating these probabilities. Due to the problem that small numbers for the example can bias statistical estimators much we propose using the following estimator, which is robust against this effect.

$$P_+^{ss} = \frac{1}{2} \left(1 + \frac{n_+^{ss}}{n^{ss}} - \frac{n_-^{ss}}{n^{ss}} \right)$$

$$P_-^{ss} = \frac{1}{2} \left(1 + \frac{n_-^{ss}}{n^{ss}} - \frac{n_+^{ss}}{n^{ss}} \right) = 1 - P_+^{ss}$$

where 2 accounts for the number of classes.

In the general case the probability of a class i is,

$$P_i^{ss} = \frac{1}{K} \left(1 + (K-1) \frac{n_i^{ss}}{n^{ss}} - \sum_{j, \forall j \neq i} \frac{n_j^{ss}}{n^{ss}} \right)$$

where K is the total number of classes and j accounts for all the classes except i .

This model consists in a function which outcome ranges in $[0, 1]$, hence with probabilistic interpretation, that not only takes into account the class examples to determine the probability of that class (positive or negative in our case), but also other classes examples. Additionally, it also takes into account the densities of examples for different classes.

Calculating the probability in this way is similar to assuming that, in lack of any evidence, each state in the subspace has the same “proportion” of a positive and a negative example (uniform distribution). This proportion will change as new evidence is gathered as a function of the density of classes. For instance, the probability for a positive example could range from 0, when all the covered states are instantiated with negative examples, to 1 when the whole subspace is occupied with positive instances.

After these definitions we can continue with the *cec* generation procedure: After calculating the probabilities for a working subspace, if the probability of a positive example is nonzero, a *cec* is created and added to the list of *cecs* that will be used for planning. Every new *cec* is composed of,

$$cec_{new} = \langle P_{new} = ss_w, a_{pcec}, O_{new} = O_{pcec} \rangle$$

where ss_w is the working subspace.

The process of *cec* generation is performed whenever a *pcec* is created and for every recently promoted working subspace that have nonzero probability for a positive example.

About Exploration

In the decision making system exploration of actions is dictated by the teacher and the planner. Nevertheless, as the planner generates plans using *cecs*, which not yet completely evaluated we use forced exploration that permits learning a more complete model of the world dynamics.

Learning Module

The learning module of the decision making system learns rules and *pcecs* evaluating under which conditions a change would occur after an action execution. Incomplete knowledge of the conditions necessary to afford changes leads to uncertainties about the occurrence of the change under a situation. Nevertheless, it is possible to estimate the probability for that change to occur given a subspace. In the next section we present how these estimations are improved, and how the minimal set of conditions that affords the expected changes is found.

Learning *pcecs*

The core of the method for learning *pcecs* is the estimation of probabilities for a positive and a negative example for different subspaces. The estimations and subspaces generation procedures are guided by experienced states stored in L_s . The aim is to find the smallest sets of conditions for which the expected change has a high probability to occur.

The learning process consists in selectively storing positive and negatives examples related to each *pcecs*, and refining the SS_w representation by promoting candidate subspaces from SS_c to SS_w .

Initialization

The generation of a *pcec* occurs after executing an instructed action. The initial structure of a *pcec* consists in the expected outcome O_{pcec} , which involves the changed conditions after the action execution, the action itself a_{pcec} =instructed action, and the initial set of working and candidate subspaces.

The initial set of working subspaces SS_w is composed by one subspace formed with the conditions changed following the action a_{pcec} . In the case of SS_c , the initial set of subspaces is composed by subspaces formed with the conditions changed with a_{pcec} , and one additional condition of a detector not involved in the changes. It is considered one candidate subspace for each condition of those detectors.

Memorizing Examples

Every experienced state, either positive or negative, is stored in L_s of the *pcec* used to create the *cec* whenever the probability of error P_e (probability of misclassification) is greater than a critic threshold P_c .

The probability of the error is calculated as [21],

$$P_e = \min[P_-^{SS_w}, P_+^{SS_w}]$$

The threshold P_c is determined using a linear function of the density of the subspace,

$$P_c = P_e^{\max} \delta = \frac{\delta}{2}$$

where δ is the density calculated as,

$$\delta = \frac{n_-^{SS_w} + n_+^{SS_w}}{n^{SS_w}}$$

and P_e^{\max} is the maximum possible probability of error.

In this way, for low density subspaces the probability error is less significant and examples are then stored to reduce the uncertainty in the estimations, while, as the density becomes larger, the probability of the error becomes more meaningful.

Note that the storing criterion implies that every *cec* should have a pointer to the *pcec*, from which it originated and to the working subspace of that *pcec*, which was used to set all its conditions.

Finally, after an example is stored, all the counters of the subspaces in SS_w and SS_c , the conditions of which are included in the stored example, are updated.

Promoting Candidate Subspaces

The necessity of improving the estimation by promoting candidate subspaces is also measured using the probability of the error and the density of the subspaces. The promotion of candidate subspaces occurs when enough examples were stored but the estimation capability of the system is still bad. To consider this requirement we propose using a threshold for the probability of the error that is a linear decreasing function of the density. In this manner, when the evaluated working subspace has a few examples the uncertainty is high and more evidence should be accumulated before promoting any candidate subspace. On the other hand, when the working subspace is densely occupied with examples, higher probabilities of the error are more trustable as indicators of the necessity for refining the representation. Then, the threshold for the probability of the error is calculated as,

$$P_c^{prom} = (P_e^{\min} - P_e^{\max})\delta + P_e^{\max}$$

where P_e^{\min} is the highest error allowed, which plays the role of an upper bound for the precision in the estimation, preventing over-fitting, and allowing a better treatment of noisy data.

After the calculation, if the probability of error of the evaluated working subspace is above the threshold, then, from all the candidate subspaces from SS_c involving the evaluated working subspace, the one with lowest probability of error is promoted. To save computational resources, only working subspaces related to *cecs* that produce surprises are evaluated.

Generating Candidate Subspaces

Every time a candidate subspace is promoted new candidate subspaces are generated.

The generation takes place for each consistent combination of the recently promoted subspace with the conditions of another working subspace of the *pcec*. All the possible combinations will produce new candidate subspaces.

Finally, once the candidate subspaces are generated, the counters of examples are initialized in accordance to the positive and negative examples covered by them.

Learning Rules

So far, we have explained how *pcecs* are learned from experience and how *cecs* are generated from *pcecs*. Now, we will explain how to relieve the job of the planner by memorizing sequences of successfully executed *cecs* into a macro planning operator called *rule*.

To guarantee successful execution of a sequence, the precondition of the to-be-executed rule should ensure the occurrence of the *cec*-preconditions in the proper order. This is

achieved firstly by accumulating (in the precondition part of the rule) the preconditions of the *cecs* needed to afford the changes. In case a detector takes more than one condition value during the sequence, the condition closer to the start of the sequence is considered as it should occur first.

On the other hand, the outcome of the rule should enumerate the results obtained after the execution of the whole sequence. This is done by accumulating the outcomes of each *cec* into the outcome-part of the rule. As latest outcomes in the sequence cancel early ones when the same detectors are involved, the rule outcome should consider for each detector the condition of those *cec* later in the sequence whenever the same detector is involved.

It is important to remark that, in this first approach, we let the teacher control the rule generation by the instruction given. The teacher will instruct a single action when no sequence is convenient to be merged into a rule, and he/she will instruct a sequence of actions whenever he/she knows that this sequence will be need many times during the task or could be difficult to find for the planner.

Cecs and Rules Correction

When a surprise arises, the *pcec* from which the *cec* was obtained is evaluated. For this we require that there is a working subspace in the *pcec* that has higher probability for being a positive example than the subspace from which the *cec* originated. Furthermore this subspace has to be consistent with the whole sequence of *cecs* stored in the rule. Only then an update is performed. This is done by replacing the conditions of the *cecs* with the conditions of the working subspace with the highest probability.

Consistency in the sequence requires that the changes produced by previous *cecs* in the sequence as well as all the preconditions of posteriors *cecs* are contemplated in the situations where a *cec* should be applied. Additionally, changed affordance may require conditions that do not change by themselves but are nonetheless necessary for the execution of the *cec*. Those conditions should also be guaranteed. Thus, to update a *cec* in a rule, all the previous restrictions need to be verified.

If the *cec* results are modified, the rule correction is simply performed by updating the rule preconditions and outcomes with the new added conditions. This is done following the procedure of rule generation but applied only to the modified parts.

Outline of the Algorithm

In this section we present the algorithm of the decision making system using a pseudo-code.

Pseudo-code

```
INIT system
Define GOAL
WHILE goal is not reached
{
  IF PLAN found
  {
    Execute PLAN
  }
  ELSE (plan not found)
  {
```

```

Teacher instructs actions
FOR each action instructed,
{
  Execute action
  GENERATE new pcec
  GENERATE new cec from new pcec
  APPEND new cec to LCECS
}
GENERATE RULES using LCECS
} (else if plan found)
}(end while GOAL is not reached)

```

Execute PLAN

```

FOR each cec in PLAN
{
  Execute action
  IF surprise
  {
    IF high uncertainty in the estimations
    {
      STORE negative example
    }
    If necessity of promoting candidate subspace
    {
      PROMOTE best candidate subspace
      GENERATE new candidate subspaces
      CORRECT cec with promoted subspace
      CORRECT rules containing cec
    }
    EXIT FOR (stop plan execution and replan)
  }
  ELSE (no surprise)
  {
    IF high uncertainty in the estimations
    {
      STORE positive example
    }
  }
} (end for)

```

The learning module procedures are detailed in figure 2. To contextualize see that these procedures are those that take place inside the learning box in the schema of figure 1.

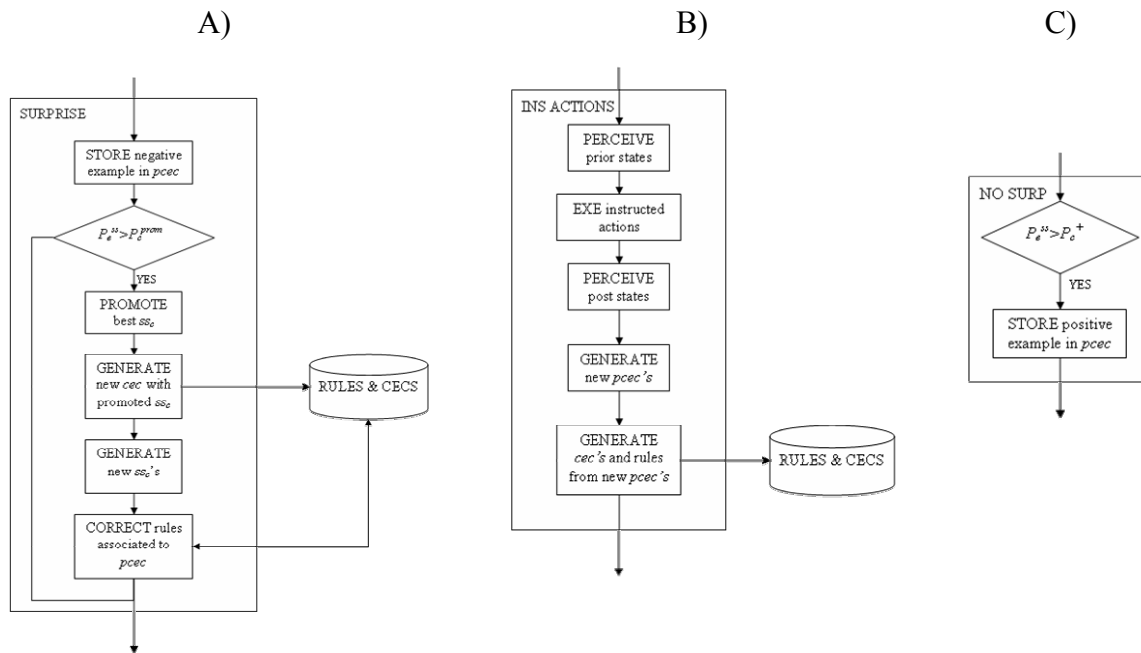


Figure 2. A) Learning when there is a surprise; B) Learning after action instructions; C) Learning after a successful execution of a *cec*.

Conclusions

The contribution of this work is an extension of the previous system presented in [5]. The general architecture of the previous system is maintained but there is a significant improvement in the process of learning cause-effects. In [5] the cause-effects were learned with the help of human explanations about relevant conditions that afford changes. Now this process is completely automatic permitting not only to learn of cause-effects without the help of the teacher but also to estimate the chances of producing the desired outcome for any set of conditions. Hence, the learning method could be now applied to stochastic and non-stationary environments. On the other hand, this probabilistic approach may constitute another step toward a possible formalization of the OAC: an object description is not restricted to a unique set of conditions but multiple set of conditions with different chances of affording the object functionality.

Nevertheless, there are still many other pending issues to treat like the evaluation of the method in real scenarios and the automatic explorations of actions whenever the planner fails to find a plan. Other topics are the definition of a criterion for plan memorization into rules and the integration of the learning method with an advanced planner module.

References

- [1] Newell, A. The knowledge level. *Artificial Intelligence*, 18(1), 87-127, 1982.
- [2] Brooks, R. "Intelligence without representation". *Artificial Intelligence*, 47, pp. 139-159, 1991.
- [3] Mitchell, T. *Machine Learning*. McGraw Hill. 1997
- [4] Piaget, J. *The origins of intelligence in children*. New York: International Universities Press. 1952.

- [5] Agostini A., Celaya E., Torras C., Wörgötter F. Action Rule Induction from Cause-Effect Pairs Learned Through Robot-Teacher Interaction. In Proc. of the International Conference on Cognitive Systems, CogSys 2008. (Karlsruhe, Germany). April 2008, pp. 213-218.
- [6] Wörgötter F., Agostini A., Krüger N., Shylo N., Porr B. Cognitive Agents - A Procedural Perspective relying on the Predictability of Object-Action-Complexes (OACs), Robotics and Autonomous Systems, 2008 (In press. doi:10.1016/j.robot.2008.06.011).
- [7] Geib, C., Mourao, K., Petrick, R., Pugeault, N., Steedman, M., Krüger, N. and Wörgötter, F.. Object Action Complexes as an Interface for Planning and Robot Control, presented at IEEE RAS Int Conf. Humanoid Robot, Genova, Italy, 2006.
- [8] <http://www.paco-plus.org>.
- [9] Lemaître, M., Verfaillie, G.. Interaction between reactive and deliberative tasks for on-line decision-making. Presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [10] Gat, E. On three-layer architectures. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors. Artificial Intelligence and Mobile Robots. MIT/AAAI Press, pp. 195-210, 1998.
- [11] Schoppers, M. J.. Universal Plans for Reactive Robots in Unpredictable Environments. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI 87), Milan, Italy, 1987, pp. 1039-1046.
- [12] Newton, M., Levine, J.. Evolving Macro-Actions for Planning. Presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [13] Nicolescu, N., Mataric, M.. A Hierarchical Architecture for Behavior-Based Robots. In Proc. of the 1st Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, Bolgna, Italy, 2002, pp. 227-233.
- [14] Rao, D., Jiang, Z., Jiang, Y.. Learning Activation Rules for Derived Predicates from Plan Examples. Presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [15] Yoon, S., Kambhampati, S. Towards Model-lite Planning: A Proposal For Learning & Planning with Incomplete Domain Models. Presented at the 2007 International Conference on Automated Planning and Scheduling, Providence, Rhode Island, USA, 2007.
- [16] Wang, X. Learning planning operators by observation and practice. In Proceedings of the Second International Conference on AI Planning Systems, Chicago, IL, USA, 1994.
- [17] Oates, T. and Cohen, P. Learning planning operators with conditional and probabilistic effects. In Proceedings of the AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems, 1996, pp. 86-94.
- [18] Benson, S. Inductive learning of reactive action models. In Proceedings of the 12th International Conference of Machine Learning, 1995, pp. 47-54.
- [19] Sutton, R. and Barto, A. Reinforcement Learning. An Introduction. MIT Press, 1998.

- [20] La Valle, S. Planning Algorithms. Cambridge University Press. 2006.
- [21] Duda, R., Hart. P and Stork, D. Pattern Recognition. John Wiley & Sons, Inc., 2nd Edition, 2001.

A HIERARCHICAL 3D CIRCLE DETECTION ALGORITHM APPLIED IN A GRASPING SCENARIO

Emre Başeski, Dirk Kraft, Norbert Krüger

The Maersk Mc-Kinney Møller Institute, University of Southern Denmark

Campusvej 55 DK-5230 Odense M, Odense, Denmark

{emre,kraft,norbert}@mmmi.sdu.dk

Keywords: 3D circle detection, grasping, stereo vision, hierarchical representation.

Abstract: In this work, we address the problem of 3D circle detection in a hierarchical representation which contains 2D and 3D information in the form of multi-modal primitives and their perceptual organizations in terms of contours. Semantic reasoning on higher levels leads to hypotheses that then become verified on lower levels by feedback mechanisms. The effects of uncertainties in visually extracted 3D information can be minimized by detecting a shape in 2D and calculating its dimensions and location in 3D. Therefore, we use the fact that the perspective projection of a circle on the image plane is an ellipse and we create 3D circle hypotheses from 2D ellipses and the planes that they lie on. Afterwards, these hypotheses are verified in 2D, where the orientation and location information is more reliable than in 3D. For evaluation purposes, the algorithm is applied in a robotics application for grasping cylindrical objects.

1 INTRODUCTION

Circles are important structures in machine vision since they are a common feature for natural and human-made objects and they provide more information than points and lines about the pose of an object. In 3D vision, there are various ways of obtaining edge-like 3D entities (sparse stereo) from a stereo camera setup. Once the sparse stereo data is grouped with respect to a perceptual organization scheme, certain structures can be extracted from individual or combinations of these perceptual groups. Both, in dense and sparse stereo the correspondence finding phase in 3D reconstruction reduces the reliability of the information. Therefore, while detecting a certain structure like a 3D circle by using this kind of information, one needs to take into account the noise and uncertainty of the information.

The algorithms that are used to detect 3D circles can be grouped into three categories. The first category consists of *voting algorithms* like the Hough transform (Duda et al., 2000). Due to the size of the parameter space, voting algorithms require much more memory and computational power than other algorithms.

The second category contains *analytical algorithms* which use the geometric properties of circles (e.g., (Xavier et al., 2005)). For laser-range data, this kind of algorithms run fast and are robust because of the high-reliability of input data. Stereo vision on the other hand, introduces too many outliers and uncertainties that make the geometrical properties unstable.

The last category involves *fitting algorithms*. They are traditionally based on minimizing a cost function which depends on a distance function that measures errors between given points and the fitted circle (Jiang and Cheng, 2005; Chernov and Lesort, 2005; Shakarji, 1998). The fitting process can be done either in 3D or in 2D. If it is done in 2D, the optimal plane for the given points is calculated and the points are projected onto that plane. If the fitting is done in 3D, the minimization starts with an initial estimate and tries to converge to the optimal circle. However, to guarantee convergence, a good initialization is required. This can be done by starting with multiple initializations, which decreases the computational efficiency drastically. One can reduce the parameter space as in (Jiang and Cheng, 2005) but the noisy nature of stereo vision data decreases the probability of convergence. Therefore, although fitting in 2D is a

decoupled solution (plane fitting and curve fitting are handled separately), it is more advantageous in terms of efficiency and reliability for noisy data.

In this article, an algorithm which is based on fitting in 2D is presented. Note that, the common practice for such approaches is using only 3D information and its projection onto 2D. The main specificity of our approach is, instead of using 3D information only, a hierarchical representation is used which represents visual information at different levels of semantic (e.g., 2D versus 3D) as well as different spatial complexity (local versus global). By that we obtain information with different levels reliability. Furthermore, there is a verification process, which is also performed using different levels in the representation hierarchy.

In this work, the hierarchical representation presented in (Krüger et al., 2004) is used. An example is presented in Figure 1 which shows what kind of information exists on different levels of the representation. At the lowest level of the hierarchy, there is the image with its pixel values (Figure 1(a)). At the second level, there exists the filtering results (Figure 1(b)) which give rise to the multi-modal 2D primitives at the third level (Figure 1(c)). At the third level, not only the 2D primitives but also 2D contours (Figure 1(d)) are available that are created using the perceptual organization scheme in (Pugeault et al., 2006). The last level contains 3D primitives and 3D contours (Figure 1(e-f)) created from 2D information of the input images.

Since the reliability and the amount of data decreases as the level of the representation hierarchy increases ((Pugeault et al., 2008)), lower levels should be used to verify the operations done in higher levels. For example, localization of a shape in 3D can be checked in 2D, once the perspective projection of the shape is known. Note that, there are more primitives and their orientation and location information is more reliable in 2D.

The key idea of our approach is to use different aspects of visual information according to their locality/globality, their semantic richness as well as their reliability in an efficient way. For example, it is known that 2D information is more reliable than 3D (since the stereo correspondence problem introduces additional errors) but 3D information is required to find 3D position, 3D orientation, and the radius of a circle. We make use of this trade-off, so that semantic reasoning on a higher level (e.g., 3D information leads to 3D hypotheses) becomes verified on a lower but more reliable level (e.g., 2D information) by feedback mechanisms. Another aspect is the locality of the data being used at the different steps of processing. By using semi-global features (i.e., 2D and 3D

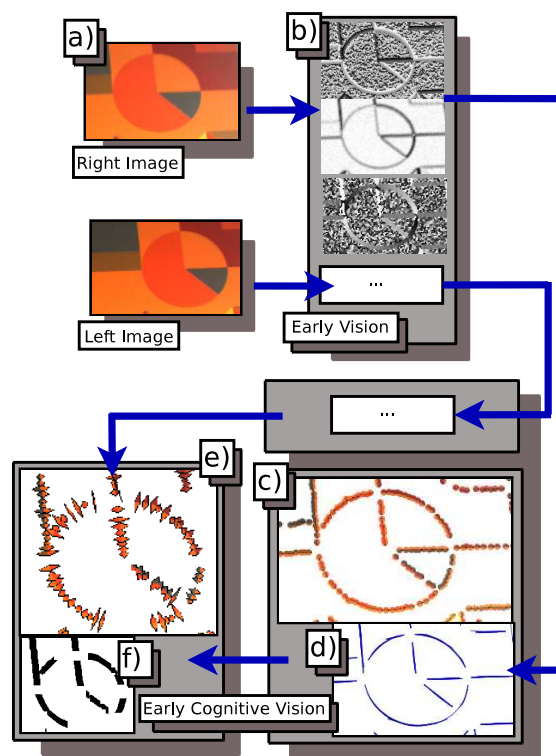


Figure 1: Different type of information that is available in the representation hierarchy (a) Original image (b) Filtering results (c) 2D primitives (d) 2D contours (e) 3D primitives (f) 3D contours.

contours) for the computation of hypotheses we decrease computational time significantly. Since these hypotheses are verified using local features, the effect of additional errors inherent in contours are minimized. In this way, we make optimal use of the different levels of the hierarchical representation.

The rest of the article is organized as follows: In Section 2, the circle detection algorithm is introduced and some evaluation results in different scenarios with high variation in terms of circle sizes, 3D positions and orientation as well as number of circles and other factors such as occlusion are discussed. The experiments done on different objects in a grasping scenario where 3D dimension and location play an important role are presented in Section 3. We conclude with an evaluation of the algorithm based on these experiments.

2 CIRCLE DETECTION

The algorithm can be summarized in four steps as (1) ellipse hypotheses creation (Section 2.1), (2) verification of these hypotheses (Section 2.2), (3) creating circles by transferring the verified hypotheses to 3D

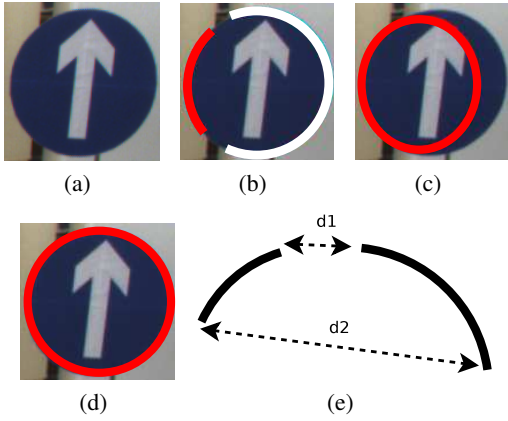


Figure 2: (a) Original image (b) Two contours on the circle (One is red and the other is white) (c) Fitted ellipse to the red contour in (b) (d) Fitted ellipse to the white contour in (b) (e) Two curves can be merged if $\min(d1, d2)$ is small enough.

(Section 2.3) and (4) verifying the created circles in 2D (Section 2.4).

2.1 Computing Ellipse Hypotheses

Because of the correspondence problem in the 3D reconstruction process, the information in 2D can not be transferred to 3D completely. Therefore, contours in 2D contain more primitives than corresponding 3D contours and a 2D contour can contain projections of more than one 3D contour. These facts are the motivation to use 2D contours to search for 2D ellipses in the image. Another important fact is that, a single 2D contour may not be big enough to compute the ellipse that we are searching for. In Figure 2(c) and (d), the ellipses fitted to contours in Figure 2(b) are shown. Since the red contour is not big enough, the ellipse fitted to that contour is not the desired one.

Having too small data sets for fitting is a common problem originating from perceptual organization. To overcome this difficulty, a merging mechanism has been proposed in (Ji and Haralick, 1999) which is based on proximity. Two curve segments are merged if the distance between their closest end points is smaller than a certain value (Figure 2(e)). The first step of the algorithm starts with merging the 2D contours by using the proximity criterion. This merging operation creates a new set of 2D contours which contain the old 2D contours and their combinations.

Let C_i be the set of all 3D contours whose projections on the image plane are contained in the 2D contour c_i . Then, for the 3D contour C_j , $P \cdot C_j \in c_i$ iff $C_j \in C_i$ (P is the projection matrix). Note that when two 2D contours are combined, the result is

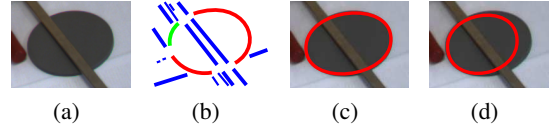


Figure 3: (a)Input image (b)2D contours (c) A true ellipse (d)A false ellipse.

represented as c_k^+ and the set of 3D contours whose projections on the image plane are contained by the combination is represented as C_k^+ .

The ellipse hypotheses e_k that the 3D circles are based on are created from the combined contours where c_k^+ is the 2D combined contour to which e_k is fitted. The ellipse fitting is done using the algorithm in (Pilu et al., 1996) which is an ellipse specific least-squares fitting method. The fitted ellipses are represented using the general ellipse equation given in (1).

$$ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0 \quad (1)$$

2.2 Verification of Ellipse Hypotheses

Since we use the merged contours, the fitting procedure creates a lot of false ellipses as well as true ones. Therefore, not all the fitted ellipses are really in the scene. A true ellipse is shown in Figure 3(c) which is fitted to the combination of the two red contours in Figure 3(b) and a false ellipse is shown in Figure 3(d) which is fitted to the combination of the bottom red and the green contour in Figure 3(b).

The elimination of false ellipses is done by finding the *significance* (Lowe, 1987) of the ellipses. The percentage of covered length of e_i is calculated from all 2D primitives (represented by π_j) that satisfy the following equations:

$$\|\pi_j - e_i\| \leq \alpha_1 \quad (2)$$

$$\left| \arctan\left(\frac{d}{dx} e_i|_{(\bar{x}_j, \bar{y}_j)}\right) - \theta_j \right| \leq \alpha_2 \quad (3)$$

where α_1 and α_2 are thresholds, (2) is the distance between π_j and e_i , (3) is the difference between the slope of e_i at (\bar{x}_j, \bar{y}_j) and the orientation of π_j (represented by θ_j) and (\bar{x}_j, \bar{y}_j) is the coordinate of the closest point on e_i to π_j . If π_j satisfies (2) and (3), its patch size (the diameter of the patch covered by the primitive) is added to the total covered length of e_i . If the percentage of total covered length of e_i with respect to its perimeter is higher than a threshold, namely α_3 , the ellipse is qualified as a true ellipse. The true ellipses for some scenes are shown in Figure 4 where $\alpha_1 = 1 \text{ pixel}$, $\alpha_2 = 10^\circ$ and $\alpha_3 = 60\%$.



Figure 4: Some true ellipse examples.

2.3 Computing 3D Circle Hypotheses

Due to the fact that the perspective projection of a circle on the image plane is an ellipse, it is possible to reconstruct the 3D circle, once the plane that the circle lies on is known. Therefore, at this point, to create 3D circles, the only further information we need is the plane p_i on which the circle that will be created from ellipse e_i lies. After calculating p_i , camera geometry can be used to find all the parameters of the 3D circle whose perspective projection is e_i . Since we know the 2D contour c_i^+ which gave rise to e_i , it is possible to use the 3D contours C_i^+ whose projections are contained by c_i^+ to fit p_i . This operation gives the normal vector of the 3D circle as it is parallel to the normal vector of p_i . What is missing for the 3D circle is the center and the radius in 3D.

To find the center and the radius of the circle, discrete points on e_i are multiplied with the pseudo-inverse of the projection matrix (P^+) to create rays, passing through the camera center and the discrete points of the ellipse. The intersections of these rays and the fitted plane p_i gives 3D points which are supposed to belong to the 3D circle. The center of mass of these 3D points gives the center of the 3D circle and this center is used to calculate the radius as the average distance of the 3D points to the center. Note that, the 3D circles calculated in the this step can be represented in parametric form as:

$$R \cos(t) \vec{u} + R \sin(t) (\vec{n} \times \vec{u}) + \vec{c} \quad (4)$$

where \vec{u} is a unit vector from the center of the circle to any point on the circumference; R is the radius; \vec{n} is a unit vector perpendicular to the plane and \vec{c} is the center of the circle.

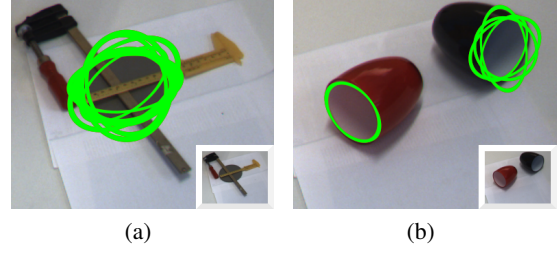


Figure 5: (a-b) Projection of 3D circles on the image plane before verification.

Some results are presented in Figure 5(a-b). Note that more than one combined contour can represent the same ellipse and they produce correct circles as well as false ones because of the 3D reconstruction uncertainties. The false circles are eliminated in the next step.

2.4 Final Selection of Circle Hypotheses

As the last step, our aim is to find which 3D circle is the best for ellipses that have been represented by more than one combined contour. Let \mathcal{E}_i be the set of ellipses that are similar. It is impossible for them to have the same curve parameters so we can measure the similarity between two ellipses as a cost function depending on the distance between their centers, the difference of their perimeters and orientations. The main idea of the last step is to calculate the *significance* of ellipses which are projections of circles created from the ellipses in set \mathcal{E}_i . We do the evaluation in 2D since the amount and the reliability of data in this dimension is higher than 3D. To find the ellipse which is the perspective projection of a 3D circle, we can pick 5 points of the circle on the image plane and use the implicit equation of the conic through 5 points as in (5).

$$\begin{vmatrix} x^2 & xy & y^2 & x & y & 1 \\ x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ & & \dots & & & \\ x_5^2 & x_5 y_5 & y_5^2 & x_5 & y_5 & 1 \end{vmatrix} = 0 \quad (5)$$

The 5 points can be created from (4) for $t \in \{0, 80 \dots 320\}$. Equation 5 gives the generic equation of an ellipse as in (1). Therefore, we find the *significance* of these projected ellipses by using all 2D primitives π_j that satisfy Equations (2) and (3). For each set \mathcal{E}_i , only the one circle with the highest significance is kept. Some results are presented in Figure 6 and 7.

2.5 Problems

Although the algorithm is stable on tilted, partially occluded and cluttered circles, perceptual organiza-

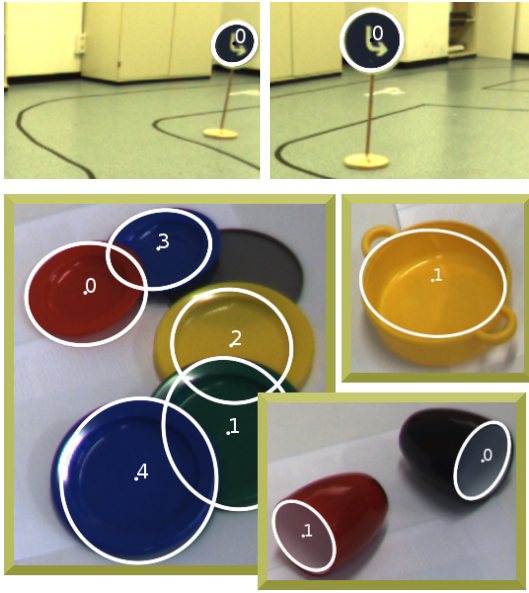


Figure 6: 3D circle detection results on different scenarios. (White ellipses are the projections of 3D circles onto the image plane).

tion can create problems in case of good continuation between circular and non-circular parts. Figure 8(b) illustrates a case, where the red 2D contour combines a circular and a non-circular part. In such cases, the remaining circular part (e.g., green contour in Figure 8(b)) may create a valid ellipse hypothesis but transferring this hypothesis to 3D is heavily dependent on the plane that is fitted to the 3D points and usually this situation leads to incorrect 3D circles as shown in Figure 8(c).

3 APPLICATION IN A GRASPING SCENARIO

The algorithm described in the previous section is applied in a robot grasping application. In this section we describe the setup and use of this application to evaluate the circle detection.

3.1 System Description

The robotic system used consist of a six degree of freedom industrial robot (Stäubli RX-60B), a two finger parallel gripper (Schunk PG 70) and a Point Grey BumbleBee2 stereo camera (see Figure 9(a)). The camera is calibrated relative to the robot coordinate system. Therefore the output of the above algorithm can be directly used for the computation of the grasping position.

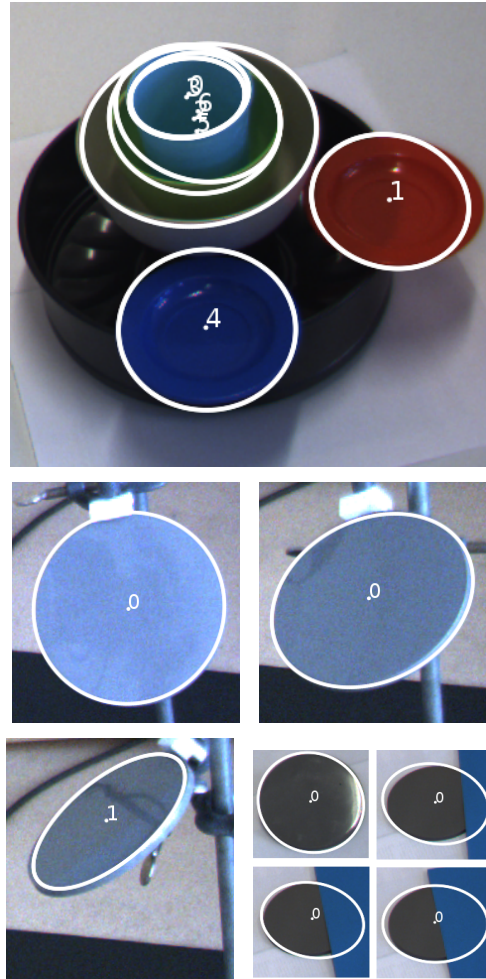


Figure 7: 3D circle detection results for multiple objects, different orientation and occlusion. (White ellipses are the projections of 3D circles onto the image plane).

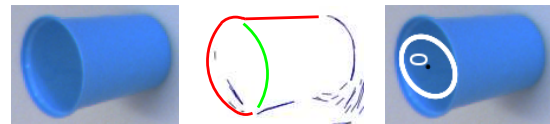
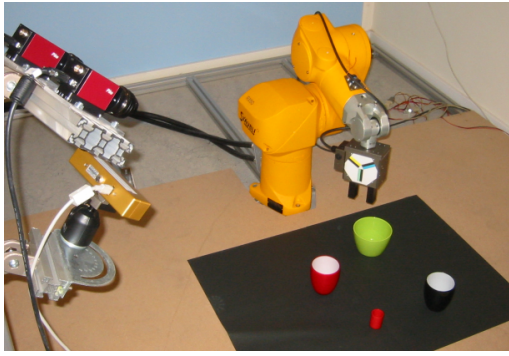


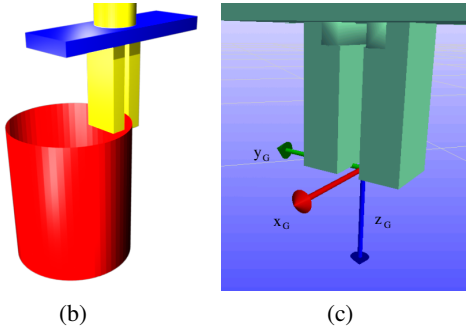
Figure 8: (a)Original image (b) 2D contours corresponding to (a) (c) Detected 3D circle.

3.2 Grasp Definition

For this work we selected one of the grasps defined in the grasping application to evaluate the quality of the circle detection. The cylindrical object is grasped on its brim (see Figure 9(b)). The position of the grasp is expressed similar to the parametric form in (4). From this observation directly follows that there is actually not one possible grasp, but a one dimensional manifold of grasps (varying the grasp position around the circumference of the circle). Additionally the grasp-



(a)



(b)

(c)

Figure 9: (a) Robot system consisting of six degree of freedom industrial robot, two finger gripper and two stereo camera systems (The lower camera systems was used for this work). (b) Grasp at the brim of the cylindrical object. (c) Gripper coordinate system.

ing depth h can be chosen according to the requirements of the scene. The position p of the grasper can therefore be defined as:

$$\vec{p} = R\cos(t)\vec{u} + R\sin(t)(\vec{n} \times \vec{u}) + \vec{c} - \vec{n}h. \quad (6)$$

Figure 9(c) shows the position and orientation of the grasper coordinate system defined at the end of the fingers. The grasper needs to be aligned in the following way: $\vec{z}_G = -\vec{n}$ and $\vec{y}_G = \cos(t)\vec{u} + \sin(t)(\vec{n} \times \vec{u})$. While the gripper opening can be defined as $d = \min(2R, d_{max})$.

3.3 Evaluation

Figure 10 shows a number of scenarios where the gripper is moved to the grasping position computed based on the circle information ($h = 2\text{ cm}$, t was used in a standard configuration except when this would have lead to a collision). For the set of experiments shown, the number of true positives (a circle that exists in the scene is detected) is 35, the number of false negatives (a circle that exists in the scene is not detected) is 1 and the number of false positives (a circle is detected that is not present in the scene) is 13. As a conclusion, 97.2% of the circles present in the

scene have been detected and out of all detected circles (true positives and false positives), 72.9% of them correspond to the circles present in the scene. Note that, the false positives occur for relatively big circles where the numerical stability decreases. On the other hand, using the saliency measure (which is high for true positives) of the found circles, the true positives have higher chance to be chosen for grasping. Also, the different setups show that our system is able to cope with different levels of complexity.

4 CONCLUSION

We have discussed a 3D circle detection algorithm which makes use of different aspects of 2D and 3D information for hypothesis generation and verification. To be able to cope with the uncertainties of sparse stereo data, 3D circles are localized in 3D by considering 2D hypotheses and verified in 2D, where the information is more reliable. The potential of the approach has been shown on a grasping application for different scenarios. As a future work, the problem of combining circular and non-circular parts will be handled by splitting 2D contours with respect to junctions and 3D structure of the contour.

ACKNOWLEDGEMENTS

The work described in this paper was conducted within the EU Cognitive Systems project PACOPLUS (IST-FP6-IP-027657) funded by the European Commission.

REFERENCES

- Chernov, N. and Lesort, C. (2005). Least Squares Fitting of Circles. *J. Math. Imaging Vis.*, 23(3):239–252.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience Publication.
- Ji, Q. and Haralick, R. M. (1999). A Statistically Efficient Method for Ellipse Detection. In *ICIP (2)*, pages 730–734.
- Jiang, X. and Cheng, D.-C. (2005). Fitting of 3D Circles and Ellipses Using a Parameter Decomposition Approach. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 103–109. IEEE Computer Society.
- Krüger, N., Lappe, M., and Wörgötter, F. (2004). Biologically Motivated Multi-modal Processing of Visual Primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428.

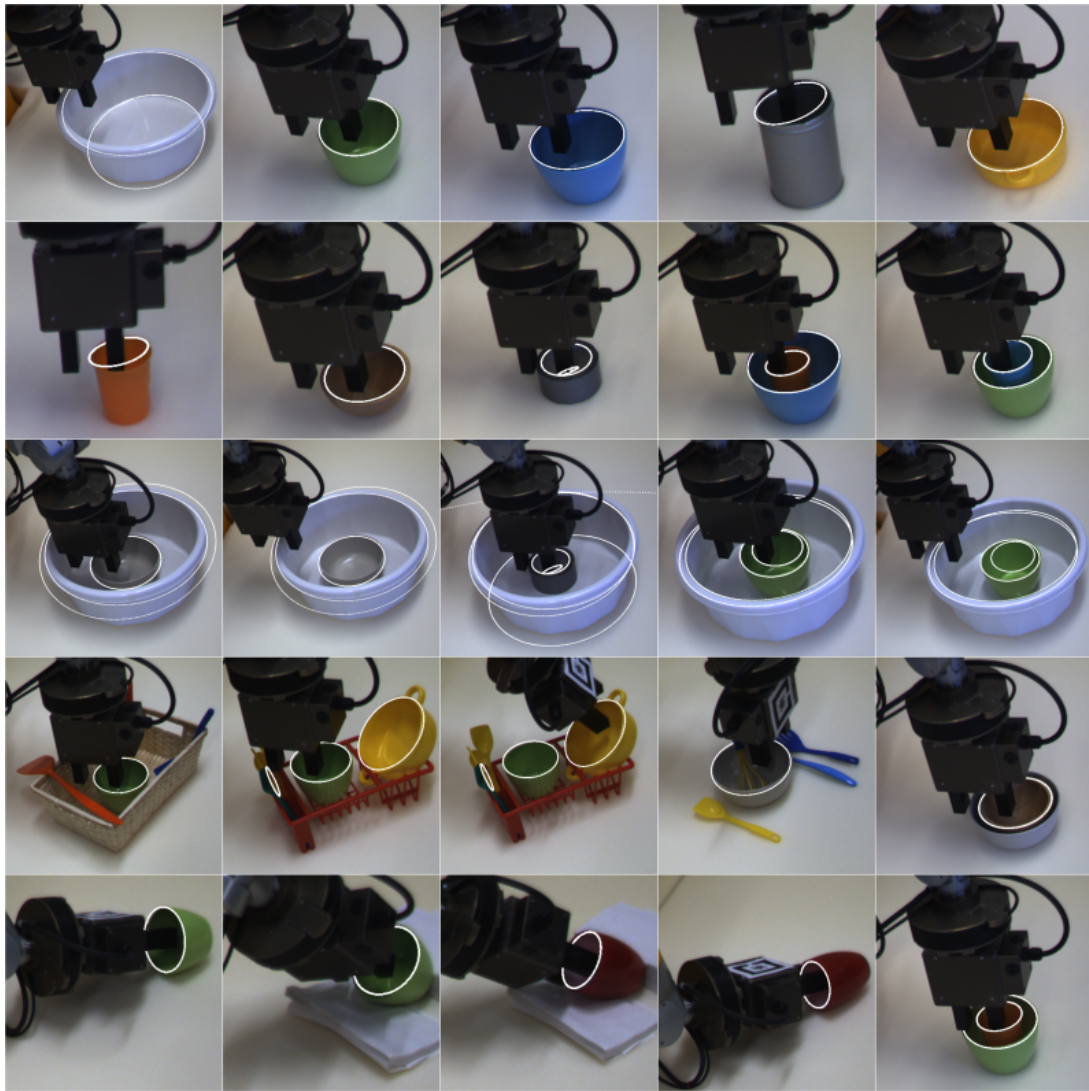


Figure 10: Detected circles and applied grasps. The circles were drawn into the images and the occluded parts were corrected afterward to improve the readers scene understanding. The scenes are of different complexity, starting out with single objects, going to objects included in each other, multiple (and more complex) objects and finally tilted single objects.

Lowe, D. G. (1987). Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31(3):355–395.

Pilu, M., Fitzgibbon, A., and Fisher, R. (1996). Ellipse-Specific Direct Least-Square Fitting. In *In Proc. IEEE ICIP*.

Pugeault, N., Kalkan, S., Başeski, E., Wörgötter, F., and Krüger, N. (2008). Reconstruction Uncertainty and 3D Relations. In *Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08)*.

Pugeault, N., Wörgötter, F., and Krüger, N. (2006). Multi-modal Scene Reconstruction Using Perceptual Grouping Constraints. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*.

Shakarji, C. (1998). Least-Squares Fitting Algorithms of

the NIST Algorithm Testing System. *Res. Nat. Inst. Stand. Techn.*, 103:633–641.

Xavier, J., Pacheco, M., Castro, D., Ruano, A., and Nunes, U. (2005). Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3930–3935.

Segment tracking via a spatiotemporal linking process including feedback stabilization in an n-d lattice model

Babette Dellen^{1,2}, Eren Erdal Aksoy³, and Florentin Wörgötter³

¹Bernstein Center for Computational Neuroscience Göttingen, Max-Planck Institute for Dynamics and Self-Organization, Germany

²Institut de Robotica i Informatica Industrial (CSIC-UPC), Barcelona, Spain

³Bernstein Center for Computational Neuroscience Göttingen, University Göttingen, Germany
Email: bkdellen@bccn-goettingen.de

Abstract

We extend the concept of spatially synchronous dynamics in spin-lattice models to the spatiotemporal domain to track segments within an image sequence. The method is related to synchronization processes in neural networks and based on superparamagnetic clustering of data. Spin interactions result in the formation of clusters of correlated spins, providing an automatic labeling of corresponding image regions. The algorithm obeys detailed balance. This is an important property as it allows for consistent spin-transfer across subsequent frames, which can be used for segment tracking. In the tracking process the correct equilibrium will, thus, always be found, which is an important advance as compared to other, more heuristic, tracking procedures. In the case of long image sequences, i.e. movies, the algorithm is augmented with a feedback mechanism, further stabilizing segment tracking.

1. Introduction

How can we make sense out of a complex visual scene having no or only little prior knowledge about its contents and the objects therein? Such problems occur for example if we wish to learn cause-effects in an hitherto unknown environment. Vice versa, many object definitions are

only meaningful within the context of a given scenario and a set of possible actions.

Object tracking, i.e. the assignment of consistent labels to objects in different frames of a video, is important for solving various tasks in the field of computer vision, including automatic surveillance, human-computer interaction, and traffic monitoring. [37]. Most object tracking algorithms require that predefined objects of interests are detected in the first frame or in every frame of the movie. In an unknown scenario however, we prefer to track image segments, presumably representing parts of objects, thus postponing object definition to a later step of the visual-scene analysis. Several approaches for segment tracking have been proposed in the context of video segmentation [7, 27, 31, 9, 21, 38, 34, 19]. Some approaches rely on segmenting each frame independently, e.g. by classifying pixels into regions based on similarity in the feature space, followed by a segment matching step based on their low-level features [7, 27, 31, 9]. Other methods use motion projection to link segments, i.e. the position of a segment in a future frame is estimated from its current position and motion features [21, 38, 34, 19].

In this paper, we group pixels based on a feature similarity criterium using a method based on the superparamagnetic clustering of data. Tracking of segments is accomplished through simultaneous segmentation of adjacent frames which are linked using local correspondence information, e.g. computed via standard algorithms for optic flow [3]. Blatt et al. (1998) first formulated the segmentation problem in terms of Potts models of granular ferromagnets or spins [22]. In the superparamagnetic phase, segments, i.e. ordered regions of aligned spins, appear naturally. To further accelerate the relaxation of the spin system, Opara and Wörgötter (1998) introduced an energy-based cluster updating technique (ECU), based on the cluster-updating method of Swendsen and Wang [28], and applied the algorithm to the problem of image segmen-

tation [20, 33].

The motivation for our choice is threefold. First, in superparamagnetic clustering the number of segments is determined by the algorithm itself and thus does not need to be predefined. Second, the method obeys detailed balance, ensuring that the algorithm converges to a stable solution independent of the initial conditions. Third, the concept of spin-relaxation can be easily extended to space-time by allowing bonds to be formed between spins belonging to different movie frames. Thus, time, i.e. frame number, just takes the role of additional dimensions in the spin-relaxation process, using only energy minimization without further constraints.

The segmentation (or partition) of an image is sensitive to global and local changes of the image, i.e. small changes in illumination, the appearance/disappearance of objects parts, causing the partition to change from one frame to the next. By synchronizing the segmentation process of adjacent frames, these kind of partitioning instabilities can be reduced. Furthermore, segment correspondences can be established without having to employ segment matching. To further stabilize segment tracking in the case of long image sequences, we developed a feedback control mechanism, which allows segmentation instabilities, e.g. sudden disappearances of segments, to be detected and removed by adjusting a control parameter of the segmentation algorithm.

The paper is structured as follows: In Section 2, we extend the method of superparamagnetic clustering in spin models to the temporal dimension and introduce the controller algorithm. In Section 3, we first verificate the core algorithm using short image sequences because these are more suitable to introduce and test the method. We further investigate the sensitivity of the algorithm to system parameters and noise. Then, we demonstrate that segment tracking can be achieved for real movies. In Section 4, the results are discussed.

2. Algorithmic framework

Segment tracking can be roughly divided into the following subtasks: (i) image segmentation, (ii) linking (tracking), and (iii) stabilization. The third point acknowledges that segments, unlike objects, are not per se stable entities, but are sensitive to changes in the visual scene. Subtasks (i-ii) will be solved using a conjoint spin-relaxation process emulated in an n-D lattice, which defines the core algorithm (Section 2.1). Local correspondence information for linking is obtained using standard algorithms for either stereo or optic flow [26, 3]. The conjoint segmentation approach has the advantage that the spin-relaxation processes of adjacent images synchronize, reducing partitioning instabilities.

Since simultaneous segmentation of long image sequences is practically impossible due to the high computational costs, we usually split the image sequence into a sequence of pairs. For example, the subsequent frames t_0 , t_1 and t_2 are split into two pairs $\{t_0, t_1\}$ and $\{t_1, t_2\}$, where the last frame of previous pair is identical to the first frame of the next pair. If a segment of the last frame of $\{t_0, t_1\}$ and a segment of the first frame of $\{t_1, t_2\}$ occupy the same image region, we can assign the same segment label to both segments. This way segments can be tracked through the entire sequence. Since the algorithm preserves detailed balance (Section 2.2), spins can be transferred from one frame to the next, greatly reducing the number of iterations required to achieve a stable segmentation.

We further stabilize segment tracking by introducing a feedback controller (Section 2.3). In long image sequences, partitioning instabilities are likely to arise at some point during the tracking process. Thus, segments may be lost due to merging or splitting of segments. The feedback

controller detects these kind of instabilities and adjust a control parameter of the core algorithm to recover the original segments.

2.1. Core algorithm

The method of superparamagnetic clustering has been previously used to segment single images [5, 20, 33]. Applying this framework to image sequences requires spin interactions to take place across frames. Due camera and object motion the images undergo changes during the course of time. To connect different frames, the mapping from one frame to the next needs to be known at least in some approximation. We solve this problem in the following way: Point correspondences, derived using algorithms for disparity or optic-flow computation, can be incorporated into the Potts model by allowing spins belonging to different frames of the image sequence to interact if the respective pixels belong to locally corresponding image points. Then, spins belonging to the different frames of the sequences are relaxed simultaneously, resulting in a synchronized segmentation of the images of the sequence. The inter-frame spin interactions cause the spins of corresponding image regions to align, and, thus, they will be assigned to the same segment. Since the formation of segments is a collective process, the point correspondences do not have to be very accurate nor does the algorithm require point correspondences for each pixel. It is usually sufficient if the available correspondences capture the characteristics of the scene only roughly.

The aim of this work is to find corresponding image regions in image sequences, i.e. stereo pairs and motion sequences. The segment tracking task is formulated as follows. Given an image sequence S containing points $p(x, y, z)$ with coordinates (x, y, z) as elements, where x

and y label the position within each image, while z labels the frame number, then we want to find a partitioning $\mathbf{P} = \mathbf{P}_1, \dots, \mathbf{P}_M$ of S in M groups such that

- (i) $\mathbf{P}_i \cap \mathbf{P}_j = \emptyset$ and $\mathbf{P}_i \neq \emptyset$ for all groups
- (ii) if point $p \in \mathbf{P}_i$, then $s(p, \mathbf{P}_i) > s(p, \mathbf{P}_j)$, where s is a function measuring the average distance of a point to the elements of a group
- (iii) if $p(x_i, y_i, z_i) \in \mathbf{P}_r$, then $p(x_i + \Delta x_i, y_i + \Delta y_i, z_i + 1) \in \mathbf{P}_r$, where Δx_i and Δy_i are the shifts of point $p(x_i, y_i, z_i)$ along the x and y axes, respectively, from frame z_i to frame $z_i + 1$.

To perform this task, we assign a spin variable σ_i (or label) to each pixel (or site) i of the image sequence. To incorporate constraints in form of local correspondence information, we distinguish between neighbors within a single frame (2D bonds) and neighbors across frame (n-D bonds). We create a 2D bond $(i, k)_{2D}$ between two pixels within the same frame with coordinates (x_i, y_i, z_i) and (x_k, y_k, z_k) if

$$|(x_i - x_k)| \leq \varepsilon_{2D} \quad (1)$$

$$|(y_i - y_k)| \leq \varepsilon_{2D} \quad (2)$$

$$z_i = z_k \quad , \quad (3)$$

where ε_{2D} is the 2D-interaction range of the spins, a parameter of the system. Across frames, we

create a n-D bond $(i, j)_{nD}$ between two spins i and j if

$$|(x_i + d_{ij}^x - x_j)| \leq \varepsilon_{nD} \quad (4)$$

$$|(y_i + d_{ij}^y - y_j)| \leq \varepsilon_{nD} \quad (5)$$

$$z_i \neq z_j \quad (6)$$

$$a_{ij} > \tau \quad , \quad (7)$$

where ε_{nD} is the n-D interaction range. The values d_{ij}^x and d_{ij}^y are the shifts of the pixels between frames z_i and z_j along the axis x and y , respectively, obtained from the optic-flow map or disparity map. The parameters a_{ij} are the respective amplitudes (or confidences), and τ is a threshold, removing all local correspondences having a small amplitude.

We define for every bond on the lattice the distance

$$\Delta_{ij} = |g_i - g_j| \quad , \quad (8)$$

where g_i and g_j are the gray (color) values of the pixels i and j , respectively. The mean distance $\bar{\Delta}$ is obtained by averaging over all bonds. We further define an interaction strength

$$J_{ij} = 1 - \Delta/\bar{\Delta} \quad . \quad (9)$$

The spin model is now implemented such a way that neighboring spins with similar color have the tendency to align. We use a q -state Potts model [22] with the Hamiltonian

$$H = - \sum_{\langle ik \rangle_{2D}} J_{ik} \delta(\sigma_i - \sigma_k) - \sum_{\langle ij \rangle_{nD}} J_{ij} \delta(\sigma_i - \sigma_j) \quad . \quad (10)$$

Here, $\langle ik \rangle_{2D}$ and $\langle ij \rangle_{nD}$ denote that i, k and i, j are connected by bonds $(i, k)_{2D}$ and $(i, j)_{nD}$, respectively. The Kronecker δ function is defined as $\delta(a) = 1$ if $a = 0$ and zero otherwise. The

segmentation problem is then solved by finding clusters of correlated spins in the low temperature equilibrium states of the Hamiltonian H . The total number M of segments is then determined by counting the computed segments. It is usually different from the total number q of spin states. Note that the local correspondences used in the algorithm to create n-D bonds are precomputed and are not altered or optimized when computing the equilibrium state. The computation of local correspondences is not the aim of this paper.

We solve this task by implementing a clustering algorithm. In a first step, “satisfied” bonds, i.e. bonds connecting spins of identical spins $\sigma_i = \sigma_j$, are identified. Then, in a second step, the satisfied bonds are “frozen” with a some probability P_{ij} . Pixels connected by frozen bonds define a cluster, which are updated by assigning to all spins inside the same clusters the same new value Swendsen and Wang [28]. In the method of superparamagnetic clustering proposed by [5] this is done independently for each cluster. In this paper, we will employ the method of energy-based cluster updating (ECU), where new values are assigned in consideration of the energy gain calculated for a neighborhood of the regarded cluster [20, 33]. A schematic of the spin system of an image sequence is depicted in Fig. 1A.

The ECU algorithm computing the equilibrium of H consists of the following steps:

1. Initialization: A spin value σ_i between 1 and q is assigned randomly to each spin i . Each spin represents a pixel of the image sequence.
2. Computing bond freezing probabilities: If two spins i and j are connected by a bond and are in the same spin state $\sigma_i = \sigma_j$, then the bond is frozen with a probability

$$P_{ij} = 1 - \exp(-0.5J_{ij}/T) \quad . \quad (11)$$

Negative probabilities are set to zero.

3. Cluster identification: Pixels which are connected by frozen bonds define a cluster. A pixel belonging to a cluster u has by definition no frozen bond to a pixel belonging to a different cluster v .
4. Cluster updating: We perform a Metropolis update [18, 28] that updates all spins of each cluster simultaneously to a common new spin value. The new spin value for a cluster c is computed considering the energy gain obtained from a cluster update to a new spin value w_k . This is done by considering the interactions of all spins in the cluster c with those outside the cluster, assuming that all spins of the cluster are updated to the new spin value w_k , giving an energy

$$E(W_k^c) = \sum_{i \in c} \left[- \sum_{\substack{\langle ij \rangle_{2D} \\ c_k \neq c_j}} \eta J_{ij} \delta(\sigma_i - \sigma_j) - \sum_{\substack{\langle ij \rangle_{nD} \\ c_k \neq c_j}} \eta a_{ij} J_{ij} \delta(\sigma_i - \sigma_j) \right] \quad (12)$$

where $\langle ik \rangle_{2D}, c_k \neq c_j$ and $\langle ij \rangle_{nD}, c_k \neq c_j$ are the noncluster neighborhoods of spin i , and W_k^c symbolizes the respective spin configuration. Here, N is the total number of pixels of the image sequence. The constant η is chosen to be 0.5.

Similar to a Gibbs sampler, the selecting probability $P(W_k^c)$ for choosing the new spin value to be w_k is given by

$$P(W_k^c) = \exp(E(W_k^c)) / \sum_{l=1}^q \exp(E(W_l^c)) \quad . \quad (13)$$

5. Iteration: The new spin states are returned to step 2 of the algorithm, and steps 2-5 are repeated, until the total number of clusters stabilizes.

6 Segments are defined as groups of correlated spins and can be extracted using a thresholding procedure. All pairs of pixels connected by a bond (i, j) with $c(\sigma_i, \sigma_j) > \theta$ are considered as friends. The function c computes the correlation of the spin states of i and j over several iterations. Then, all mutual friends are assigned to the same segment. Finally, M is determined by counting the total number of segments. In practice, we find it sufficient to take the clusters found in the last iteration as segments.

2.2. Detailed balance

In an earlier study we had provided evidence that this algorithm obeys detailed balance. The full proof shall not be repeated here and can be found in [20], but we will outline its idea briefly again as the existence of detailed balance is of central importance for being able to transfer spin configurations across frames.

Detailed balance assures that the proposed algorithm computes an equilibrium spin configuration, i.e. the segmentation, which minimizes the energy function on the labels, and that this is Boltzmann distributed. In more formal terms: We have two sets of variables: the spin-value configuration $W \in \Sigma$, where Σ is the space of all configurations, and (similar to the Swendsen and Wang algorithm [28]), and the cluster configuration $C \in \Gamma$, where Γ is the space of all clusters. The complete system assumes configurations in the shared configuration space $\Gamma \times \Sigma$.

The goal of the ECU algorithm is to label an image according to the energy function on the labels $E(W)$, which leads to an equilibrium probability distribution $P(W) = \exp[-E(W)/T]/Z$, where Z is the partition function. Labeling could, for example, be done simply by Gibbs sampling, but Gibbs sampling of individual spins can be very slow. To speed up sampling, we define an energy function over additional variables, the clusters c , such that the equilibrium distribution $P(W, C) = \exp([-E(W)/T]/Z)$ still has the same marginal distribution, $\sum_C P(W, C) = P(W)$, as defined above. Then we define a Markov process over this joint system consisting of two steps: (1) sampling of clusters given the spins $P(W, C \rightarrow W, C')$ and (2) sampling of spins given the clusters $P(W, C \rightarrow W', C)$. The claim to prove consists of two aspects. If detailed balance holds, applying these two steps in succession should (1) result in the desired equilibrium distribution $P(W, C)$, which has the desired marginal distribution over spins $P(W)$ and (2) it needs to be the Boltzmann distribution [28, 4].

The consequence of detailed balance is that spin states can be transferred across image pairs, where spins are being calculated for one pair (the first pair) and then pixels in the next two frames (the second pair) are just assigned these spins from where on a new relaxation process starts (see Fig. 5 for an example). Hence, the relaxation process for the second pair (and all to follow) is much faster when using spin transfer and the system will always arrive at the correct final thermodynamic equilibrium making spin-transfer based segmentation concordant across frames. Note, this property allows consistent segment tracking across many frames without additional assumptions (see Fig. 5), which requires more effort with other methods. This makes this algorithm a possibly very useful and fast enough tool for model free segment tracking applications as will be shown by one example in Fig. 7.

2.3. Feedback control

Segmentation instabilities arising during the tracking process can be partly removed by adjusting the temperature parameter of the core algorithm. The temperature choice affects the formation of segments, hence, a segment which has been lost in a previous frame, can sometimes be recovered by increasing the temperature for certain period.

The feedback controller tracks the size of the segments and reacts if the size of a segment changes suddenly. The first controller function

$$P_C^j(t) = \begin{cases} 1 & \text{if } \Delta S_j(t) < \tau_1, \\ \exp[-\Delta S_j(t)/\alpha] / \beta & \text{otherwise,} \end{cases} \quad (14)$$

measures the probability of change of segment j , where $S_j(t)$ is the size of segment j at frame t and $\Delta S_j(t) = S_j(t) - S_j(t-1)$, and α, β, τ_1 are parameters. The history of segment j in terms of occurrence is captured by the second controller function

$$P_H^j(t) = 0.4H_j(t-1) + 0.3H_j(t-2) + 0.2H_j(t-3) + 0.1H_j(t-4) \quad , \quad (15)$$

with $H_j(t) = 1$ if $S_j(t) > 0$ and zero otherwise.

Segmentation instabilities may cause a segment to be lost, for example through segment merging or splitting. We define two threshold parameters τ_2 and τ_3 . An unexpected segment loss is detected by the controller if the conditions

$$S_j = 0 \quad , \quad (16)$$

$$P_C < \tau_2 \quad , \quad (17)$$

$$\text{and } P_H > \tau_3 \quad (18)$$

are fulfilled. An unexpected segment appearance is detected by the controller if the conditions

$$P_C < \tau_2 \quad (19)$$

$$\text{and } P_H < \tau_3 \quad (20)$$

are fulfilled. The identities of the affected segments are stored by the controller. The temperature of the core algorithm is varied using predefined temperature steps ΔT . The segmentation is repeated at the new temperature $T + \Delta T$ for the affected frames. If the lost segments can be recovered at one of these temperatures, the affected segments are relabeled accordingly.

A schematic of the entire system, i.e. core algorithm with feedback control, is presented in Fig. 1B.

3. Results

We apply the algorithm to various synthetic and real image sequences. Unless otherwise indicated, the following parameter values $q = 10$, $\epsilon_{2D} = 1$, and $\epsilon_{nD} = 0$ are used. In Section 3.1, the core algorithm is applied to short image sequences and a sensitivity analysis is performed. Then, in Section 3.2, feedback control is added and the algorithm is applied to movies.

3.1. Verification of core algorithm

We first use stereo image pairs and a three-frame motion sequence to test and verificate the core method (Section 2.1) before applying the algorithm to long image sequences, i.e. movies. Stereo images are more suitable to illustrate the basic properties of the algorithm. In the following figures we use spin states instead of cluster labels to limit the total number of colors in the color-coded images to a maximum value of q . Please note that the spin states are not identical to

the cluster labels. Spins which belong to the same clusters are always in the same spin state, however, the reverse is not always true. The spin states have to be observed over several iterations to identify clusters as groups of correlated spins.

3.1.1. *Illustrating example: artificial solid square*

We first demonstrate the algorithm for a synthetic scene which contains a single, solid square, which is shifted by a disparity value of 40 pixels along the x axis, resulting in an image sequence containing two frames, labeled left and right (see Fig. 2A, left and right panel). Each image is of size 100×100 pixels². We estimate the disparity of the pixels by applying a stereo algorithm [26], which returns a disparity map D and an amplitude map A , shown in Fig. 2B-C, respectively. The disparities and the respective amplitudes determine whether a pixel in the right frame is a neighbor of a pixel in the left frame. Clustering is performed with $T = 0.01$. The spin states of the spin system are initialized to randomly chosen discrete values between 1 and 10, as depicted in Fig 2D, left and right panel. Then, the system is evolved using the energy-based cluster update algorithm described in the previous section. The spin states after 2, 5, 9, 17, 33, 65 iterations of the algorithm are shown in Fig. 2E. The process of cluster formation can be easily followed through the iterations. At iteration step 65, the pixels belonging to the square in both the left and the right frame have been assigned to the same cluster, despite incomplete disparity information.

3.1.2. *Sensitivity analysis*

We investigate the sensitivity of the algorithm in dependence of the parameter T for different levels of Gaussian white noise that we add to the solid-square stereo pair. In Fig. 3A, the ratio of the averaged number of clusters after 100 iterations, computed from 10 runs of the algorithm,

and the total number of pixels is plotted as a function of the temperature T for four different realizations of Gaussian noise with standard deviations from the absolute gray-value difference of object and background of 0%, 1%, 10%, and 20%, depicted in red, blue, black, and green, respectively. For this sequence, a perfect segmentation is achieved for $N_c = 2$, corresponding to $N_c/N = 10^{-4}$. For a noise level of 0%, the performance of the algorithm is only weakly sensitive to changes in temperature (red line). However, when adding noise to the images, the algorithm becomes more sensitive to changes in temperature (blue line), but fast saturates for increasing noise levels (black and green line). For each noise level, the segmentation results are depicted for $T = 0.3$. To establish the 3D neighborhood of image pixels here, the ground-truth disparity map of the image pair was used. However, usually, when adding noise, the quality of the disparity map decreases. Consequently, we also investigated the performance of the algorithm when computing the disparity map with a phase-based stereo algorithm. In Fig. 3B, the ratio of the number of clusters and the total number of image pixels is depicted as a function of the noise level at temperature $T = 0.1$ (black line). The ratios when using the ground truth disparity is plotted for comparison (black line). In this example, the performance is independent of the quality of the disparity map.

We further investigate the performance of the algorithm with respect to establishing correspondences on the example of the Cones stereo pair (URL: vision.middlebury.edu/stereo/). The left frame of the Cones stereo pair is shown as an inset of Fig. 3C. The percentage of wrongly assigned image points was computed independently for every segment, and the average percentage of wrongly assigned image points was plotted as a function of the mean length of the segments (Fig. 3C). A segment of length l contains l^2 image points. The plot demonstrates that the per-

formance of the algorithm is higher for large segments than for small segments, confirming our expectation that color segmentation works best for large uniform image regions. In textured areas, corresponding to very small segment sizes, the performance of the algorithm decreases rapidly.

We also investigated the influence of errors in the precomputed disparity on the performance of the algorithm by replacing disparity values of the ground-truth map randomly by erroneous values ranging from 0 to n , where n is the width of the image. In Fig. 3D the total percentage of wrongly assigned image points (taken from all segments) is plotted as a function of the density of erroneous disparity values. As expected, the performance decreases with increasing error in the disparity map. In summary: one finds that the errors are in general small and the error curves flat for larger segments corresponding to non-textured regions. It is evident that all gray (color) difference based segmentation algorithms in general do not capture textured regions and the increasing errors for small segments reflect this situation. On the other hand, it is very assuring that those segments, which follow from larger consistent gray (color) value similarities, are indeed only little affected by errors in the (stereo-)correspondence map.

3.1.3. Real stereo pair

This stereo pair shows two views of a scene of cluttered objects, i.e., paper boxes, a trash can, and a white Styrofoam object (Fig. 4A, left and right panel). Each image is of size 180×380 pixels². This stereo pair is demanding because of the amount of occlusion, the light reflexions, shadows, and the large disparities, which lead to perspective distortions, posing a problem to approaches based on segment matching. The stereo algorithm returns reliable disparity values at

the edges (Fig. 4B-C). Otherwise, the amplitude is zero (Fig. 4C). However, when performing clustering with $T = 0.2$, the algorithm is still able to segment most of the boxes into their composite surfaces (Fig. 4D-E). Some of the surfaces are partly shattered though, due to light reflexions and shadows, breaking the uniformity of the surfaces. Both the spin states after 150 and 176 iterations are shown to allow easier identification of correlated spins through visual inspection.

3.1.4. Real motion sequence

So far we had been validating our method using synthetic and real stereo pairs. Now we demonstrate that spatiotemporal synchronization of spins enables segments to be tracked through the frames of real movies.

We apply the core algorithm to three frames of a motion sequence showing a woman walking from the right to the left. The frames are of sizes 118×158 pixels² (Fig. 5A). To compute optic flow, various algorithms can be used, i.e. a differential technique by Lucas and Kanade [17]. The performance of the segmentation is only weakly sensitive to the quality of the optic-flow estimation. The optic-flow fields, coding the mapping from the frame t_0 to frame t_1 , and from frame t_1 to frame t_2 , are depicted in Fig. 5B. The spin states after 100 iterations are shown in Fig. 5C. The algorithm successfully segmented the legs, the arms, a part of the head, and parts of the background, which thus can be tracked from frame to frame. For the highly textured area in the background, no stable 3D clusters could form since the gray-value similarity of neighboring pixels is too low. However, texture could be in principal treated by performing segmentation based on texture similarity instead of color similarity.

When analyzing long motion sequences, it is inefficient to apply the algorithm to all frames at once because the computational costs increase with the number of pixels. Hence, we split the sequences in pairs of two frames at a time, where the last frame of the previous sequence is identical with the first frame of the next sequence. Then, we initialize the spin states of each sequence with the final spin states of the previous sequence. The spin states for the first sequence containing frame t_0 and t_1 after 100 iterations are shown in Fig. 5D. Then, the algorithm is applied to the next pair, containing frame t_1 and t_2 , where the spin states of both frame have been initialized to the final spin states of frame t_1 of the previous sequence. The spin states after 13 iterations are shown in Fig. 5E, demonstrating that the number of iterations required to achieve a satisfying segmentation result is greatly reduced by this technique. The number of clusters for the first sequence and the second sequence are displayed as a function of the iteration number in Fig. 5F, dashed and solid line, respectively. The number of clusters for the second sequence is plotted as a function of the iteration number at a different scale (Fig. 5G). Initially, the number of clusters decreases slightly and then approaches a stable state. In a motion sequence, the number of clusters is expected not to change much from one frame to the next. Mainly the boundaries of the clusters reorganize during the first iterations.

The segments of adjacent image pairs are connected as follows. Two segments belonging to the segmentation of frame t_1 of pair $\{t_0, t_1\}$ and frame t_1 of pair $\{t_1, t_2\}$, respectively, are assigned the same label if they occupy the same region in image frame t_1 . This way we can track the segment through the whole sequence.

3.2. Segment tracking with feedback stabilization

We add feedback control (see Section 2.3) with parameters $\alpha = 200$ pixels, $\beta = 0.8$, $\tau_1 = 50$ pixels, $\tau_2 = 0.9$, and $\tau_3 = 0.6$ to the core algorithm with temperature $T = 0.1$ and apply the algorithm to long image sequences. The first movie shows a hand taking a red apple from a plate with several fruits. A few frames of the movie are depicted in the upper panel of Fig. 6A. If the core algorithm is applied at constant temperature without feedback control, the red segment and the light pink segment, representing the respective parts of the red apple and the orange, are lost at frame number 45 due a segmentation instability: The red segment and the light pink segment merge and form a new segment, colored in light blue (see Fig. 6A, middle panel). If feedback control is included, this segmentation instability is detected and the original segments can be recovered by increasing the temperature in steps of $\Delta T = 0.1$. As a consequence, the segments can be continuously tracked, as shown in Fig. 6A (lower panel). The segments representing the cup could be recovered using the same mechanism.

The work of the feedback controller is further illustrated in Fig. 6B, where the segment size is plotted as a function of the frame number for the segments representing the red apple and the orange without and with feedback control, depicted as red, blue, brown and green lines, respectively. At frame number 45 the segment sizes of the red apple and the orange drop unexpectedly to zero (red and blue lines), thus indicating a segmentation instability (see Section 2.3). As a consequence, the feedback controller is activated and the temperature of the core algorithm is increased until the original segments are recovered (brown and green lines). The results for the whole movie are shown in Fig. 7A.

We further applied the algorithm to another movie, showing the filling of a cup with sugar

(Fig. 7B). The movie is challenging because it contains light reflexions and changing shadows. However, the algorithm is capable of tracking the main segments of the movie, i.e. the two cups and the hand.

4. Discussion

We presented an algorithm for segment tracking based on a novel, conjoint framework, combining local correspondences and image segmentation to synchronize the segmentation of adjacent images. The algorithm provides a partitioning of the image sequence in segments, such that points in a segment are more similar to each other than to points in another segment, and such that corresponding image points belong to the same segment. We tested the method on various synthetic and real image sequences, and showed stable and reliable results overall, thus fulfilling the most important requirement of segmentation algorithms. The method leads to the formation of stable region correspondences despite largely incomplete disparity or optic-flow maps. Similar algorithms for the extraction of region correspondences could potentially be constructed using other image segmentation algorithms, i.e. methods based on agglomerative clustering [35, 11]. We decided to use physics-based model for its conceptual simplicity which allowed us to integrate local correspondence information in a straightforward way. It further has the advantage that the interacting parts are inherently converging to the equilibrium state and thus are not being trapped in local extrema (detailed balance). As a consequence, the result is independent of the initial conditions, allowing us to apply the algorithm to long image sequences via spin-states transfer. This allows for consistent segment tracking across many frames without additional assumptions, which is most of the time not immediately possible with other methods. In addition,

no assumptions about the underlying data are required, e.g. the number of segments, leading to a model-free segmentation. This has the consequence that a single pixel of distinct gray value (compared to its neighbors) might define a single segment. In algorithms, which partition the image into a fixed and usually small number of segments, this phenomenon does not occur. This, however, is a problem as in all realistic situations one never knows how many segments exist and self-adjustment of the total number of segments is, thus, usually desired as compared to a pre-defined maximal number.

We further introduced a feedback controller which allows to detect segmentation instabilities, i.e. merging and splitting of segments. The feedback controller adjusts the control parameter of the core algorithm in order to recover the original segments. This allows to keep track of segment even in long movies.

Segment tracking has been performed previously in the context of video segmentation [7, 27, 31, 9, 21, 38, 34, 19]. Our method differs from these approaches in the choice of the segmentation algorithm, the way linking is achieved, and the addition of a feedback controller which detects segmentation instabilities. Superparamagnetic clustering allows a model-free unsupervised segmentation of the image sequences, including a self-adjustment of the total number of segments. Linking is introduced through local correspondence information which synchronizes the spin-relaxation process of adjacent images. This approach has the advantage that the partitioning of adjacent images are less prone to partitioning instabilities. Further, our method does not require corresponding region to fulfill any segment similarity measure. Finally, feedback control allows segmentation instabilities occurring in long sequences to be removed by assuming that “good” segments change their size in a continuous manner.

There have been a few other approaches combining image segmentation with correspondence information. The work by Toshev et al. [30] uses a joint-image graph containing edges representing intra-image similarities and inter-image feature matches to compute matching regions. Joint segmentation has also been employed by Rother et al. [24] using histogram matching.

Vision problems have been formulated in terms of energy minimization in many ways before. The major challenge of these approaches lies in the computation of the global minimum, which is often difficult in particular for interesting energy functions. Various techniques have been proposed, such as variational methods [15], graph cuts [14, 10, 25, 16, 6, 32, 36, 12], dynamic programming [1], simulated annealing [13], or relaxation labeling [8, 23, 29]. Superparamagnetic clustering has been shown to equilibrate to a global minimum for the Potts model used in this work [20]. The work of Barbu and Zhu [2], which computes disparities by minimizing energy functions through an inference algorithm defined on graph partitions, shows some similarities to the core algorithm proposed in this paper, even though it has been applied to a different problem, i.e. stereo matching. However, the algorithm of Barbu and Zhu (2005) the number of segments is a parameter to the algorithm. Unlike in superparamagnetic clustering, it is assumed that there is a natural set of labels (disparities), and a data penalty function, which makes some pixel-label assignments more likely than others. These assumptions will lead to a violation of detailed balance and spin-transfer is not possible in this framework.

In the future, we aim to track segment in unknown scenarios, for example in a robot exploration task, and to infer cause-effects from the spatiotemporal relationships of segments. We are currently working on a parallel implementation of the algorithm on GPUs to achieve real-time segment tracking for robot applications.

Acknowledgment

We thank Sinan Kalkan for valuable discussion. The work has received support from BMBF funded BCCN Göttingen, the EU Project DRIVSCO under Contract No. 016276-2, and the EU Project PACO-PLUS under Contract No. 027657.

References

- [1] Amini, A., Weymouth, T., Jain, R., 1990. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (9), 855–867.
- [2] Barbu, A., Zhu, S. C., 2005. Generalizing swendson-wang to sampling arbitrary posterior probabilities. *Trans. Pattern Anal. Mach. Intell.*, 1239–1253.
- [3] Barron, J. L., Fleet, D. J., Beauchemin, S., Burkitt, T., 1994. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 43–77.
- [4] Binder, K., Heermann, D. W., 1988. Monte carlo simulation in statistical physics. Springer Verlag.
- [5] Blatt, M., Wiseman, S., Domany, E., 1996. Superparametric clustering of data. *Physical Review Letters* 76 (18).
- [6] Boykov, Y., Veksler, O., Zabih, R., 1998. Markov random fields with efficient approximations. *IEEE Conference on Computer Vision and Pattern Recognition*, 648–655.

- [7] Choi, J. G., Lee, S.-W., Kim, S.-D., 1997. Spatio-temporal video segmentation using a joint similarity measure. *IEEE Trans. Circuits Syst. Video Technol.* 7, 279–286.
- [8] Chou, P., Brown, C., 1990. The theory and practice of bayesian image labeling. *International Journal of Computer Vision* 4 (3), 185–210.
- [9] Deng, Y., Manjunath, B. S., 2001. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Machine Intell.* 23, 800–810.
- [10] Ferrari, P., Frigessi, A., de Sa, P., 1995. Fast approximate maximum a posteriori restoration of multicolour images. *Journal of the Royal Statistical Society B* 57 (3), 485–500.
- [11] Fränti, P., Virtajoki, O., Hautamaäki, V., 2006. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (11), 1875–1881.
- [12] Gdalyahu, Y., Weinshall, D., Werman, M., 2001. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (10).
- [13] Geman, D., Geman, S., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741.
- [14] Greig, D., Porteous, B., Seheult, A., 1989. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society B* 31 (2), 271–279.

- [15] Horn, B. K. P., Schunk, B., 1981. Determining optical flow. *Artificial Intelligence* 17, 185–203.
- [16] Ishikawa, H., Geiger, D., 1998. Occlusions, discontinuities, and epipolar lines in stereo. *European Conference on Computer Vision*, 232–248.
- [17] Lucas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. *Proc. DARPA IU Workshop*, 121–130.
- [18] Metropolis, N., Rosenbluth, A. W., M. N. Rosenbluth, A. H. T., Teller, E., 1953. Equations of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087–1091.
- [19] Mezaris, V., Kompatsiaris, I., Srinivas, M. G., 2004. Video object segmentation using bayes-based temporal tracking and trajectory-based region merging. *IEEE Trans. Circuits Syst. Video Technol.* 14 (6), 782–795.
- [20] Opar, R., Wörgötter, F., 1998. A fast and robust cluster update algorithm for image segmentation in spin-lattice models without annealing – visual latencies revisited. *Neural Computation* 10, 1547–1566.
- [21] Patras, L., Hendriks, E. A., Lagendijk, R. L., 2001. Video segmentation by map labeling of watershed segments. *IEEE Trans. Pattern Anal. Machine Intell.* 23, 326–332.
- [22] Potts, R. B., 1952. Some generalized order-disorder transformations. *Proc. Cambridge Philos. Soc.* 48, 106–109.

- [23] Rosenfeld, A., Hummel, R. A., Zucker, S. W., 1976. Scene labeling by relaxations operations. *IEEE Transactions on Systems, Man, and Cybernetics* 6 (6), 420–433.
- [24] Rother, C., Minka, T., Blake, A., Kolmogorov, V., 2006. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In: *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, pp. 993–1000.
- [25] Roy, S., Cox, I., 1998. A maximum-flow formulation of the n-camera stereo correspondence problem.
- [26] Sabatini, S. P., Gastaldi, G., Solari, F., Diaz, J., Ros, E., Pauwels, K., Hulle, K. M. M. V., Pugeault, N., Krüger, N., 2007. Compact and accurate early vision processing in the harmonic space. *Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, Barcelona.
- [27] Salembier, P., Marques, F., 1999. Region-based representations of image and video: Segmentation tools for multimedia services. *IEEE Trans. Circuits Syst. Video Technol.* 9, 1147–1169.
- [28] Swendsen, R., Wang, S., 1987. Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters* 76 (18), 86–88.
- [29] Szeliski, R. S., 1990. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision* 5 (3), 271–302.

- [30] Toshev, A., Shi, J., Daniilidis, K., 2007. Image matching via saliency region correspondences. IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis.
- [31] Tuncel, E., Onural, L., 2000. Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-d affine motion modeling. IEEE Trans. Circuits Syst. Video Technol. 10, 776–781.
- [32] Veksler, O., 2000. Image segmentation by nested cuts. IEEE Conference on Computer Vision and Pattern Recognition 1, 339–344.
- [33] von Ferber, C., Wörgötter, F., 2000. Cluster update algorithm and recognition. Physical Review E 62, 1461–1664.
- [34] Wang, D., 1998. Unsupervised video segmentation based on watersheds and temporal tracking. IEEE Transl Circuits Syst. Video Technol. 8, 539–546.
- [35] Ward, J. H., 1963. Hierarchical grouping to optimize and objective function. J. Am. Statistical Assoc. 58, 236–244.
- [36] Wu, Z., Leahy, R., 1993. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11), 1101–1113.
- [37] Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: a survey. ACM Comput. Surv. 38 (4), 1–45.

- [38] Yokoyama, Y., Miyamoto, Y., Ohta, M., 1995. Very low bit rate video coding using arbitrarily shaped region-based motion compensation. *IEEE Trans. Circuits Syst. Video Technol.* 5, 500–507.

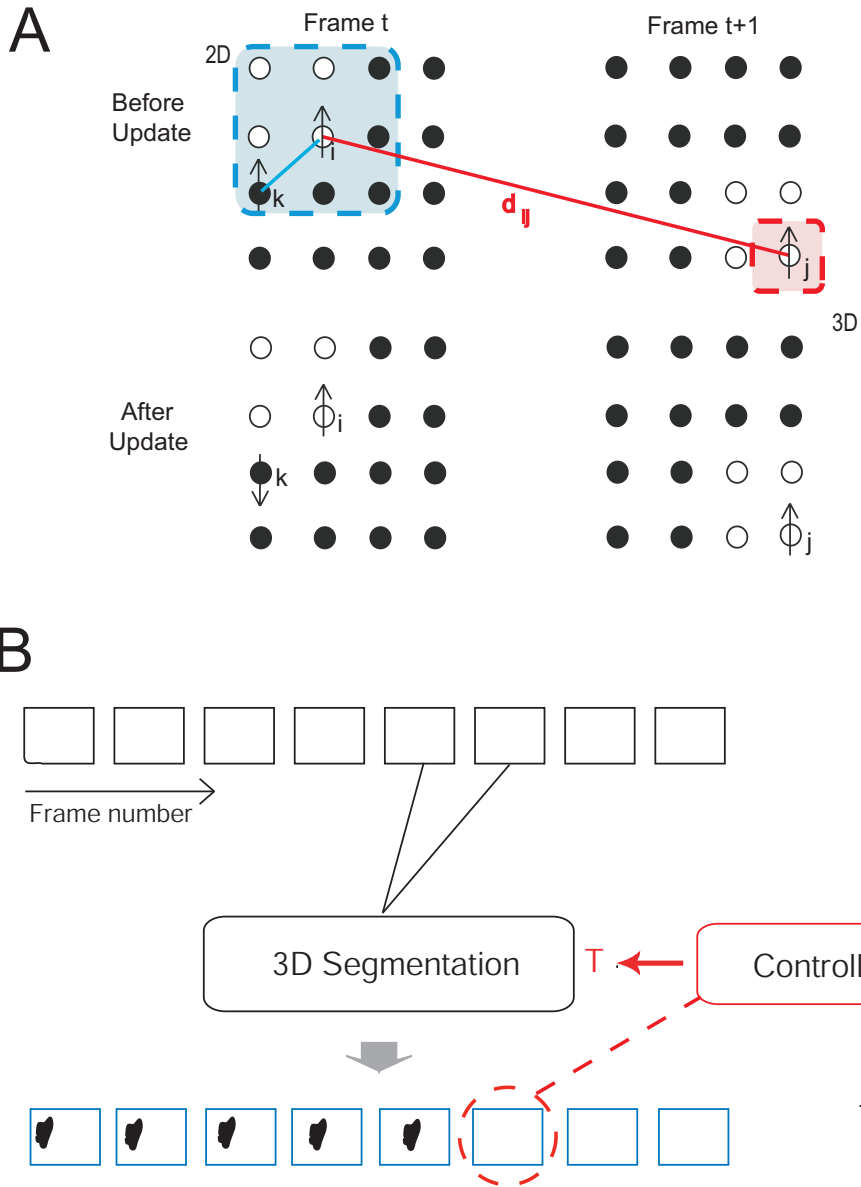


Figure 1: **A** The spin states (upward and downward pointing arrows) of pixels i , k , and j are shown before and after a spin update for two adjacent frames t and $t+1$ of an image sequence. The white and black circles indicate pixels of small and large gray values, respectively. Pixel i interacts with pixels k and j in its 2D and 3D neighborhood (shaded areas), respectively, which are in the same spin state. **B** Pairwise segmentation of movies. A feedback controller detects segmentation instabilities and adjusts the control parameter T of the core algorithm (3D segmentation) to recover lost segments.

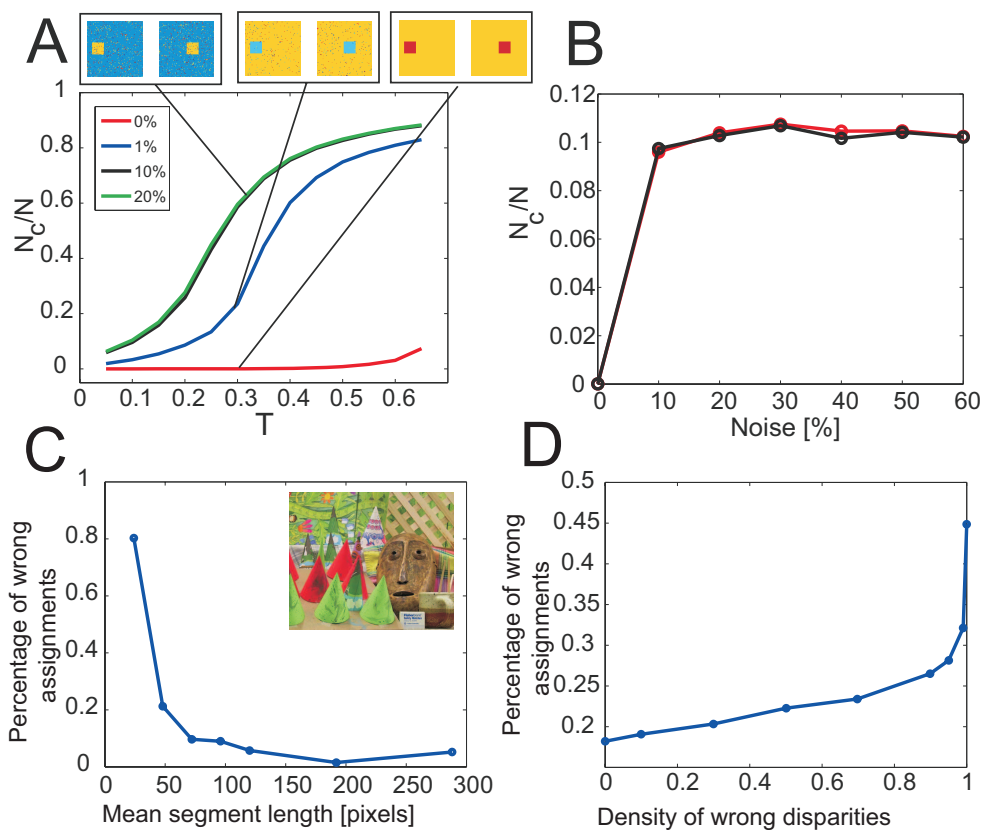


Figure 3: Sensitivity analysis. **A** The ratio of the total number of clusters N_c divided by the total number of pixels N is plotted as a function of the parameter T for different realizations of Gaussian noise, having standard deviations from the absolute gray-value difference of object and background of 0%, 1%, 10%, and 20%. The segmentation results are shown for $T = 0.3$ and different noise levels. **B** The ratio N_c/N is plotted as function of the noise level using the ground-truth disparity map (red line) and the disparity computed with a phase-based stereo algorithm [26] (black line). **C** Percentage of wrongly assigned image points as a function of the mean segment length for the Cones stereo pair (left image see inset). **D** Total percentage of wrongly assigned image points as a function of the density of erroneous disparities.

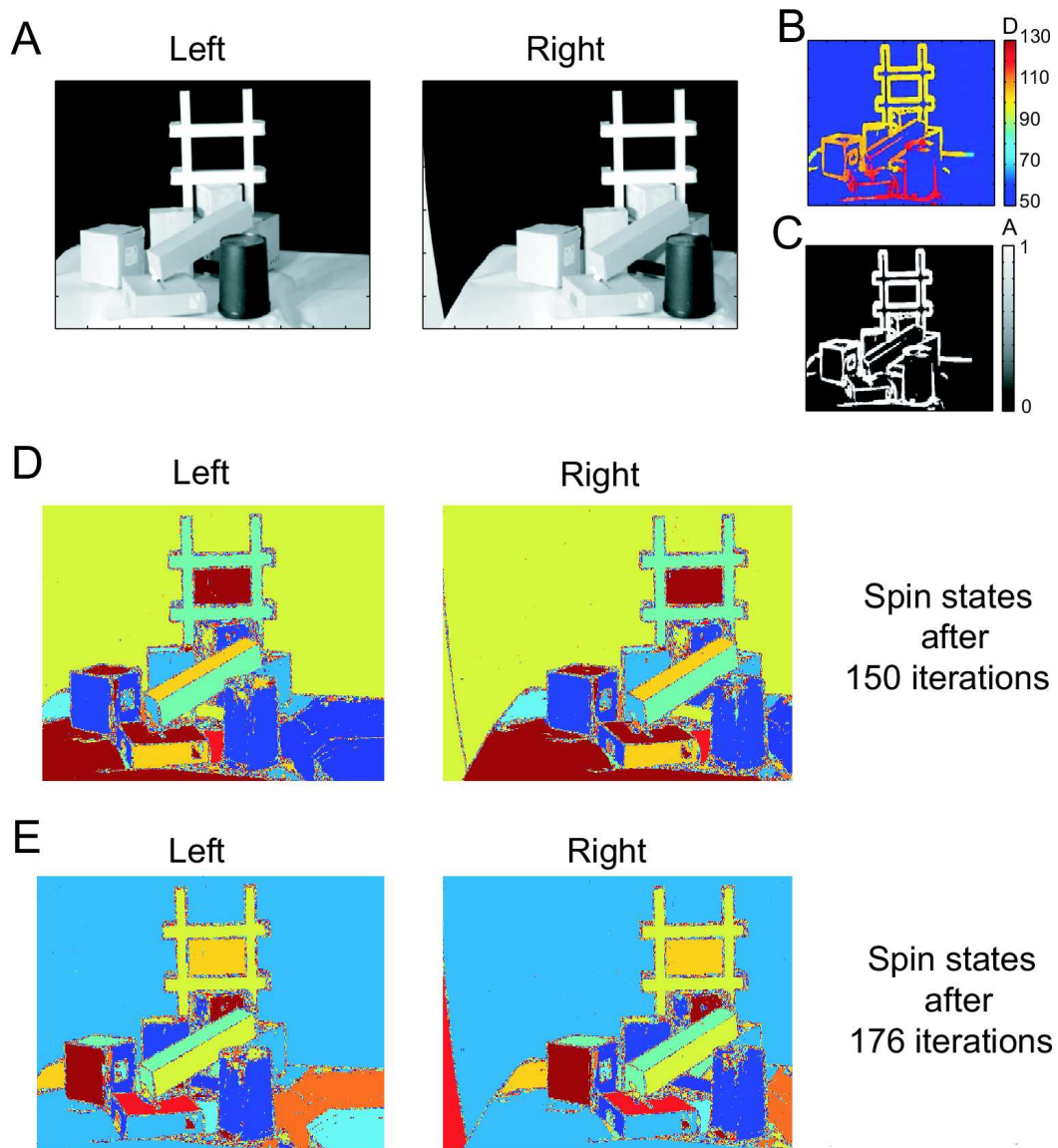


Figure 4: Cluttered-objects stereo pair. **A** Stereo image pair showing a cluttered scene containing a variety of objects. **B-C** The dense stereo algorithm returns mainly disparity information the edges of the objects. **D-E** The spin states computed by the clustering algorithm are shown after both 150 and 176 iterations for easier visual identification of the segments.

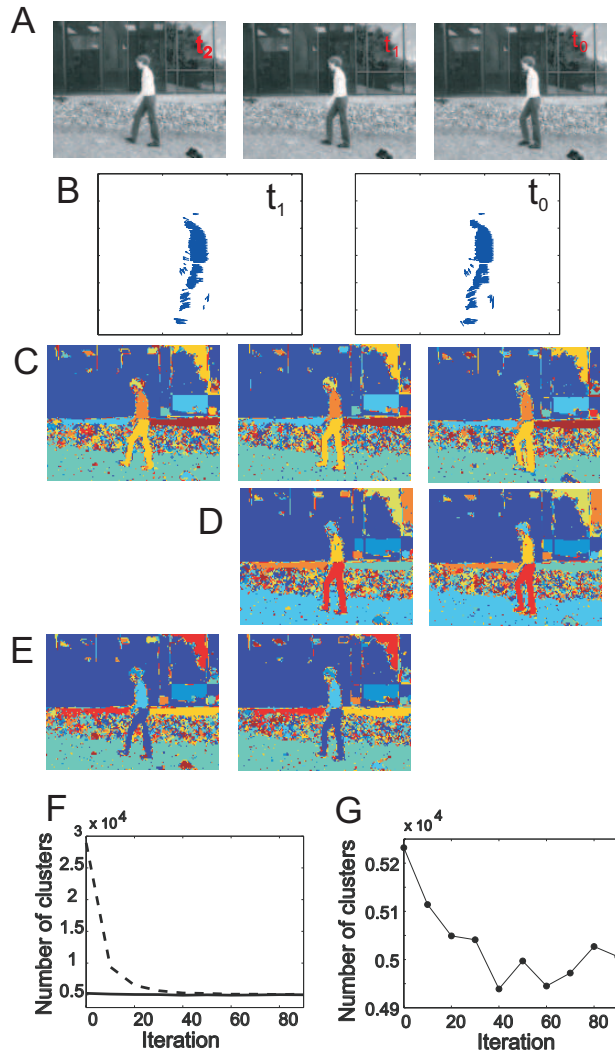


Figure 5: **A** Three frames of a motion sequence of a person walking from the right to the left, labeled t_0 , t_1 , and t_2 , respectively. **B** Estimated optic-flow fields coding the mapping from frame t_0 to frame t_1 , and from frame t_1 to frame t_2 . **C** Spin states after 100 iterations. **D** Spin states for a sequence only containing the first two frames, t_0 and t_1 . **E** Spin states for a sequence containing only the last two frames, t_1 and t_2 , where the spin states are initialized to the spin states of the previous computation. **F** The number of clusters is plotted as a function of the iteration number for the first sequence containing frame t_0 and t_1 (dashed line) and for the second sequence containing frame t_1 and t_2 (solid line). **G** Enlarged plot of the second sequence.

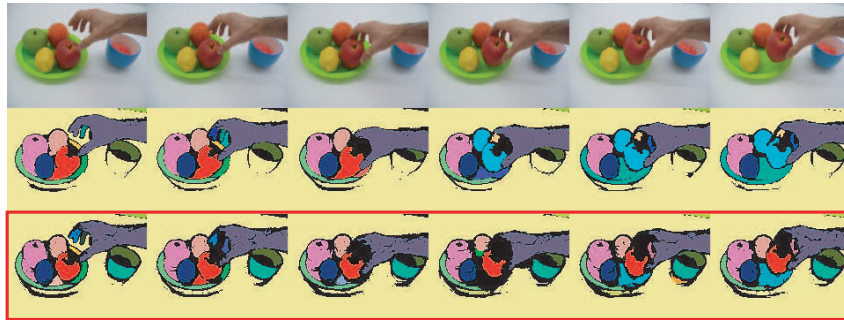
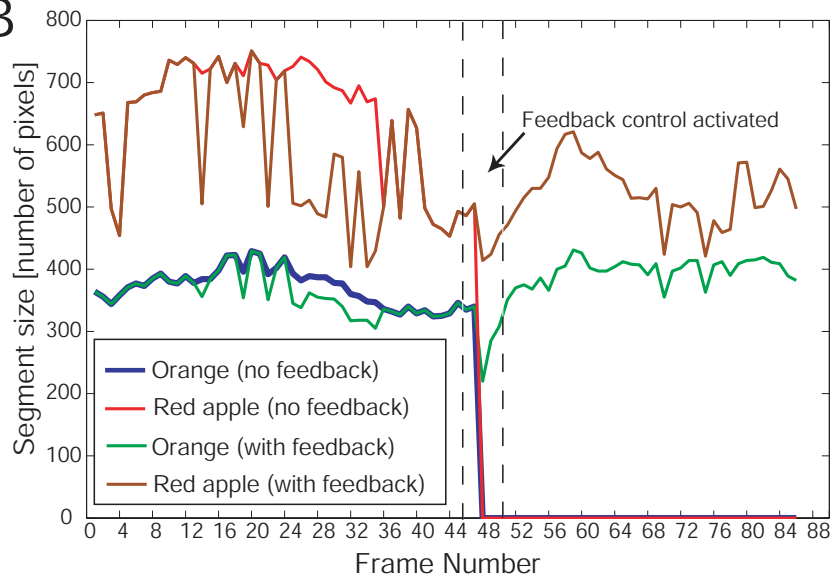
A**B**

Figure 6: Feedback control for segmentation stabilization. **A** A few frames of a movie showing a hand taking a red apple from a plate are shown together with the results of the core algorithm without and with feedback control (upper, middle, and lower panel, respectively). **B** The segment size is plotted as a function of the frame number for the segments representing the red apple and the orange without and with feedback control, depicted as red, blue, brown and green lines, respectively. At frame number 45 the segment sizes of the red apple and the orange drop unexpectedly to zero (red and blue lines), and the feedback control is activated, increasing the temperature T until the original segments are recovered (brown and green lines).

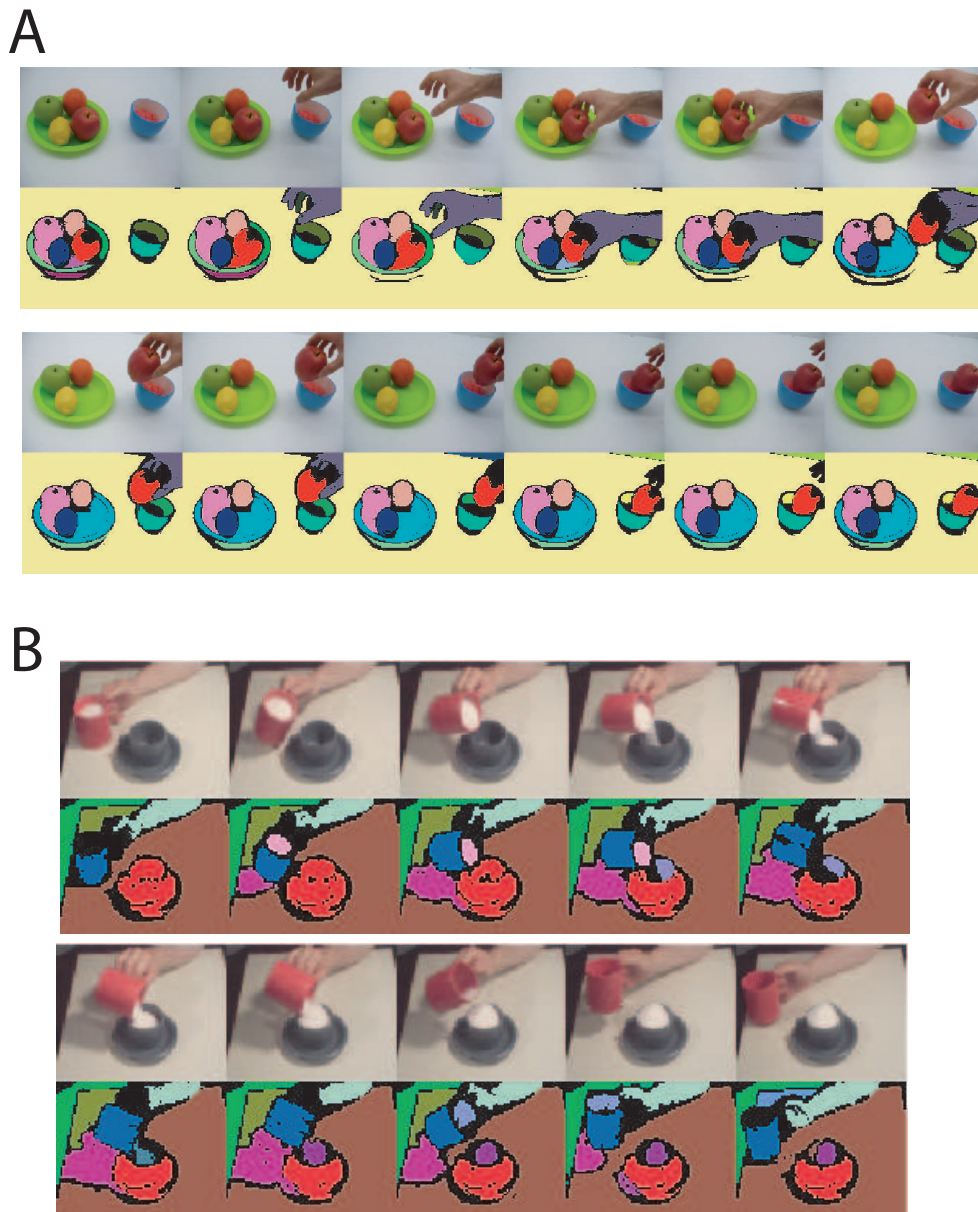


Figure 7: Segment tracking for real movies. **A** The algorithm (with feedback control) is applied to a movie showing a hand taking an apple from a plate (upper panel). The corresponding segment-tracking results are depicted below. **B** The results of the algorithm (with feedback control) for a movie showing the filling of a cup are shown.

Autonomous Learning of Object-specific Grasp Affordance Densities

R. Detry*, E. Bašeski†, M. Popović†, Y. Touati†, N. Krüger†, O. Kroemer‡, J. Peters‡ and J. Piater*

*University of Liège, Belgium. Email: Renaud.Detry@ULg.ac.be

†University of Southern Denmark.

‡MPI for Biological Cybernetics, Tübingen, Germany.

Abstract—This paper addresses the issue of learning and representing *object grasp affordances*, i.e. object-gripper relative configurations that lead to successful grasps. The purpose of grasp affordances is to organize and store the whole knowledge that an agent has about the grasping of an object, in order to facilitate reasoning on grasping solutions and their achievability. The affordance representation consists in a continuous probability density function defined on the 6D gripper pose space – 3D position and orientation –, within an object-relative reference frame. Grasp affordances are initially learned from various sources, e.g. from imitation or from visual cues, leading to *grasp hypothesis densities*. Grasp densities are attached to a learned 3D visual object model, and pose estimation of the visual model allows a robotic agent to execute *samples* from a grasp hypothesis density under various object poses. Grasp outcomes are used to learn *grasp empirical densities*, i.e. grasps that have been confirmed through experience. We show the result of learning grasp hypothesis densities from both imitation and visual cues, and present grasp empirical densities learned from physical experience by a robot.

I. INTRODUCTION

Grasping previously unknown objects is a fundamental skill of autonomous agents. Human grasping skills improve with growing experience with certain objects. In this paper, we describe a mechanism that allows a robot to learn grasp affordances of objects described by learned visual models. Our first aim is to organize and memorize, independently of grasp information sources, the whole knowledge that an agent has about the grasping of an object, in order to facilitate reasoning on grasping solutions and their likelihood of success. We represent the affordance of an object for a certain grasp type through a continuous probability density function defined on the 6D gripper pose space $SE(3)$, within an object-relative reference frame. The computational encoding is *nonparametric*: A density is represented by a large number of weighted samples called *particles*. The probabilistic density in a region of space is given by the local density of the particles in that region. The underlying continuous density function is accessed through *kernel density estimation* [21].

The second contribution of this paper is a framework that allows an agent to learn initial affordances from various grasp cues, and enrich its grasping knowledge through experience. Affordances are initially constructed from human demonstration, or from a model-based method [1]. The grasp data produced by these *grasp sources* is used to build continuous *grasp hypothesis densities* (Section VI). These densities are

attached to a 3D visual object model learned before-hand [7], which allows a robotic agent to execute *samples* from a grasp hypothesis density under arbitrary object poses, by using the visual model to estimate the 3D pose of the object.

The success rate of grasp samples depends on the source that is used to produce initial grasp data. However, no existing method can claim to be perfect. For example, data collected from imitation will suffer from the physical and mechanical difference between a human hand and a robotic gripper. In the case of grasps computed from a 3D model, results will be impeded by errors in the model, such as missing parts or imprecise geometry. In all cases, only a fraction of the hypothesis density samples will succeed; it thus seems necessary to also learn from experience. To this end, we use samples from grasp hypothesis densities that lead to a successful grasp to learn *grasp empirical densities*, i.e. grasps that have been confirmed through experience.

A unified representation of grasp affordances can potentially lead to many different applications. For instance, a grasp planner could combine a grasp density with hardware physical capabilities (robot reachability) and external constraints (obstacles) in order to select the grasp that has the largest chance of success within the subset of achievable grasps. Another possibility is the use of continuous grasp success likelihoods to infer robustness requirements on the execution particular grasp: if a grasp is centered on a narrow peak, pose estimation and servoing should be performed with more caution than when the grasp is placed in a wide region of high success likelihood.

II. RELATED WORK

Object grasps can emerge in many different ways. A popular approach is to compute grasping solutions from the geometric properties of an object, typically obtained from a 3D object model. The most popular 3D model for grasping is probably the 3D mesh [13], [17], obtained e.g. from CAD or *superquadrics fitting* [2]. However, grasping has also successfully been achieved using models consisting of 3D surface patches [20], 3D edge segments [1], or 3D points [11].

When combined with an object pose estimation technique, the previous methods allow a robot to execute a grasp on a specific object. This involves object pose estimation, computation of a grasp on the aligned model, then servoing to the object and performing the grasp [13].

Means of representing grasp affordances probabilistically have been discussed in the work of de Granville et al. [5], which is quite closely related in spirit to ours. In this work, affordances correspond to object-relative hand approach orientations, although an extension where object-relative positions are also modeled is under way [4]. The aim of the authors is to build compact sets of canonical grasp approaches from human demonstration; they mean to compress a large number of examples provided by a human teacher into a small number of clusters. An affordance is expressed through a density represented as a mixture of position-orientation kernels; machine learning techniques are used to compute mixture and kernel parameters that best fit the data. This is quite different from our approach, where a density is represented with a much larger number of simpler kernels. Conceptually, using a larger number of kernels allows us to use significantly simpler learning methods (down to mere resampling of input data, see Section VI-A). Also, the representation of a grasp cluster through a single position-orientation kernel requires the assumption that hand position and orientation are independent within the cluster, which is generally not true. Representing a cluster with many particles can intrinsically capture more of the position-orientation correlation (see Section VII, and in particular Fig. 7). The affordance densities presented by de Granville et al. correspond to the hypothesis densities developed in this paper.

III. SYSTEM OVERVIEW

The visual object model to which affordances are attached is the part-based model of Detry et al. [7] (Section IV-C). An object is modeled with a hierarchy of increasingly expressive object parts called *features*. The single top feature of a hierarchy represents the whole object. Features at the bottom of the hierarchy represent short 3D edge segments for which evidence is collected from stereo imagery via the Early-Cognitive-Vision (ECV) system of Krüger et al. [14], [19] (Section IV-A). In the following, we refer to these edge segments as *ECV descriptors*. The hierarchical model grounds its visual evidence in ECV reconstructions: a model is learned from segmented ECV descriptors, and the model can be used to recover the pose of the object within an ECV representation of a cluttered scene.

The mathematical representation of grasp densities and their association to hierarchical object models is discussed in Section V. In Section VI, we demonstrate the learning and refining of grasp densities from two grasp sources. The first source is imitation of human grasps. The second source uses a model-based algorithm which extracts grasping cues from an ECV reconstruction (Section IV-B).

IV. METHODS

A. Early Cognitive Vision

ECV descriptors [14], [19] represent short edge segments in 3D space, each ECV descriptor corresponding to a patch of about 25 square millimeters of *object* surface. To create an ECV reconstruction, pixel patches are extracted along

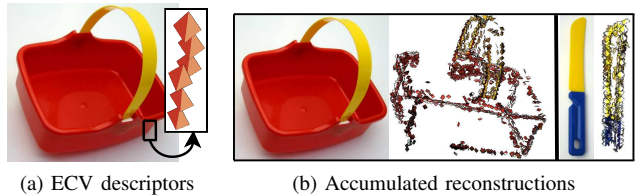


Fig. 1. ECV reconstructions

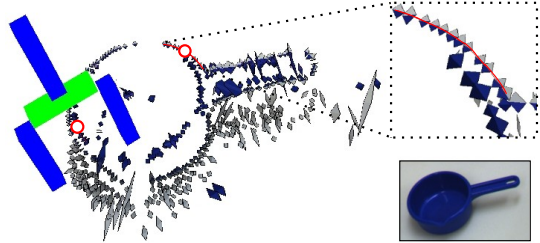


Fig. 2. Grasp reflex based on visual data.

image contours, within images captured with a calibrated stereo camera. The ECV descriptors are then computed with stereopsis across image pairs; each descriptor is thus defined by a 3D position and orientation. Descriptors may be tagged with color information, extracted from their corresponding 2D patches (Fig. 1a).

ECV reconstructions can further be improved by manipulating objects with a robot arm, and *accumulating* visual information across several views through structure-from-motion techniques [10]. Assuming that the motion adequately spans the object pose space, a complete 3D reconstruction of the object can be generated, eliminating self-occlusion issues [12] (see Fig. 1b).

B. Grasp Reflex From Co-planar ECV Descriptors

Pairs of ECV descriptors that are on the same plane and which have color information such that two similar colors are pointing towards each other can be used to define grasps. Grasp position is defined by the location of one of the descriptors. Grasp orientation is calculated from the normal of the plane linking the two descriptors, and the orientation of the descriptor at which the grasp is located [12] (see Fig. 2). The grasps generated by this method will be referred to as *reflexes*. Since each pair of co-planar descriptors generates multiple reflexes, a large number of these are available.

C. Feature Hierarchies For 3D Visual Object Representation

As explained in Section IV-A, an ECV reconstruction models a scene or an object with low-level descriptors. This section outlines a higher-level 3D object model [7] that grounds its visual evidence in ECV representations.

An object is modeled with a hierarchy of increasingly expressive object parts called *features*. Features at the bottom of the hierarchy (*primitive* features) represent ECV descriptors. Higher-level features (*meta*-features) represent geometric configurations of more elementary features. The single top feature of a hierarchy represents the object.

Unlike many part-based models, a hierarchy consists of features that may have several *instances* in a scene. To illustrate this, let us consider a part-based model of a bike, in which we assume a representation of wheels. Traditional part-based models [9], [3] would work by creating two wheel parts – one for each wheel. Our hierarchy however uses a single *generic* wheel feature; it stores the information on the existence of *two* wheels *within* the wheel feature. Likewise, a primitive feature represents a *generic* ECV descriptor, e.g. any descriptor that has a red-like color. While an object like the basket of Fig. 1 produces hundreds of red ECV descriptors, a hierarchy representing the basket will, in its simplest form, contain a single red-like primitive feature; it will encode internally that this feature has many instances within a basket object.

A hierarchy is implemented in a Markov tree. Features correspond to hidden nodes of the network; when a model is associated to a scene (during learning or detection), the pose distribution of feature i in the scene is represented through a random variable X_i . Random variables are thus defined over the pose space, which exactly corresponds to the Special Euclidean group $SE(3) = \mathbb{R}^3 \times SO(3)$. The random variable X_i associated to feature i effectively links that feature to its instances: X_i represents as one probability density function the pose distribution of all the instances of feature i , therefore avoiding specific model-to-scene correspondences.

The geometric relationship between two neighboring features i and j is encoded in a compatibility potential $\psi_{ij}(X_i, X_j)$. A compatibility potential represents the pose distribution of all the instances of the child feature in a reference frame defined by the parent feature; potentials are thus also defined on $SE(3)$.

The only observable features are primitive features, which receive evidence from the ECV system. Each primitive feature i is linked to an observed variable Y_i ; the statistical dependency between a hidden variable X_i and its observed variable Y_i is encoded in an observation potential $\phi_i(X_i)$, which represents the pose distribution of ECV descriptors that have a color similar to the color of primitive feature i .

Density functions (random variables, compatibility potentials, observation potentials) are represented nonparametrically: a density is represented by a set of particles [7].

D. Pose Estimation

The hierarchical model presented above can be used to estimate the pose of a known object in a cluttered scene. Estimating the pose of an object amounts to deriving a posterior pose density for the top feature of its hierarchy, which involves two operations [7]:

- 1) Extract ECV descriptors, and transform them into observation potentials.
- 2) Propagate evidence through the graph using an applicable inference algorithm.

Each observation potential $\phi_i(X_i)$ is built from a subset of the early-vision observations. The subset that serves to build the potential $\phi_i(X_i)$ is the subset of ECV descriptors that have a

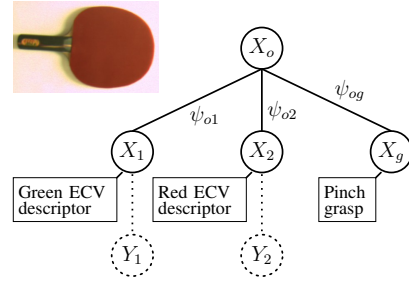


Fig. 3. Multi-sensory modeling of a table-tennis paddle with a 2-level hierarchy. The paddle is represented by feature o (top). Feature 1 represents a generic green ECV descriptor. The rectangular configuration of green edges around the handle of the paddle is encoded in ψ_{o1} . Y_1 and Y_2 are observed variables, which link features 1 and 2 to the visual evidence produced by ECV. X_g is a grasp feature, linked to the object feature through the pinch grasp affordance ψ_{og} .

color that is close enough to the color associated to primitive feature i .

Evidence is propagated through the hierarchy using a belief propagation (BP) algorithm [18], [22]. BP works by exchanging *messages* between neighboring nodes. Each message carries the belief that the sending node has about the pose of the receiving node. In other words, a message allows the sending feature to probabilistically vote for all the poses of the receiving feature that are consistent with its own pose – consistency being defined by the compatibility potential through which the message flows. Through message passing, BP propagates collected evidence from primitive features to the top of the hierarchy; each feature probabilistically votes for all possible object configurations consistent with its pose density. A consensus emerges among the available evidence, leading to one or more consistent scene interpretations. The pose likelihood for the whole object is eventually read out of the top feature; if the object is present twice in a scene, the top feature density should present two major modes. The global belief about the object pose may also be propagated from the top node down the hierarchy, reinforcing globally consistent evidence and permitting the inference of occluded features.

Algorithms that build hierarchies from accumulated ECV reconstructions are discussed in prior work [6].

V. REPRESENTING GRASP DENSITIES

This section is focused on the probabilistic representation of grasp affordances, and on the integration of grasp affordances within the hierarchical object model. By *grasp affordance*, we refer to the different ways to place a hand or a gripper near an object so that closing the gripper will produce a stable grip. The grasps we consider are parametrized by a 6D gripper pose composed of a 3D position and a 3D orientation.

A. Grasp Features

Within our framework, a grasp affordance is represented with a probability density function defined on $SE(3)$ in an object-relative reference frame. Probabilistically speaking, we store an expression of the joint distribution $\mathbf{P}(X_o, X_g)$, where X_o is the pose distribution of the object, and X_g is the grasp

affordance. This is done by adding a new “grasp” feature to the hierarchical Markov network, and linking it to the top feature (see Fig. 3). The statistical dependency of X_o and X_g is held in a compatibility potential $\psi_{og}(X_o, X_g)$, which exactly corresponds to the grasp density: $\psi_{og}(X_o, X_g)$ holds the relative configuration of grasp affordance and object pose, i.e. the grasp distribution into the reference frame of the top feature.

When an object model has been visually aligned to an object instance (i.e. when the marginal posterior of the top feature has been computed from visually-grounded bottom-up inference), the grasp affordance of the object *instance* is computed through top-down BP inference, by sending a message from X_o to X_g through $\psi_{og}(X_o, X_g)$. Intuitively, this corresponds to transforming the grasp density to align it to the current object pose, yet explicitly taking the uncertainty on object pose into account to produce a posterior grasp density that acknowledges visual noise.

B. Continuous Grasp Densities

From a mathematical point of view, grasp potentials are identical to visual potentials. They can thus be encoded with the same nonparametric density representation. Density evaluation is performed by assigning a kernel function to each particle supporting the density, and summing the evaluation of all kernels. Sampling from a distribution is performed by sampling from the kernel of a particle ℓ selected from $\mathbf{p}(\ell = i) \propto w^i$, where w^i is the weight of particle i .

Grasp densities (grasp potentials and grasp random variables) are defined on the Special Euclidean group $SE(3) = \mathbb{R}^3 \times SO(3)$, where $SO(3)$ is the Special Orthogonal group (the group of 3D rotations). We use a kernel that factorizes into two functions defined on \mathbb{R}^3 and $SO(3)$. Denoting the separation of an $SE(3)$ point x into a translation λ and a rotation θ by

$$x = (\lambda, \theta), \quad \mu = (\mu_t, \mu_r), \quad \sigma = (\sigma_t, \sigma_r),$$

we define our kernel with

$$\mathbf{K}(x; \mu, \sigma) = \mathbf{N}(\lambda; \mu_t, \sigma_t) \cdot \Theta(\theta; \mu_r, \sigma_r) \quad (1)$$

where μ is the kernel mean point, σ is the kernel bandwidth, $\mathbf{N}(\cdot)$ is a trivariate isotropic Gaussian kernel, and $\Theta(\cdot)$ is an orientation kernel defined on $SO(3)$. Denoting by θ' and μ_r' the quaternion representations of θ and μ_r [15], we define the orientation kernel with the Dimroth-Watson distribution [16]

$$\Theta(\theta; \mu_r, \sigma_r) = \mathbf{W}(\theta'; \mu_r', \sigma_r) = C_w(\sigma_r) \cdot e^{\sigma_r (\mu_r'^T \theta')^2} \quad (2)$$

where $C_w(\sigma_r)$ is a normalizing factor. This kernel corresponds to a Gaussian-like distribution on $SO(3)$. The Dimroth-Watson distribution inherently handles the double cover of $SO(3)$ by quaternions [5].

The bandwidth σ associated to a density should ideally be selected jointly in \mathbb{R}^3 and $SO(3)$. However, this is difficult to do. Instead, we set the orientation bandwidth σ_r to a constant allowing about 10° of deviation; the location bandwidth σ_t is then selected using a k -nearest neighbor technique [21].

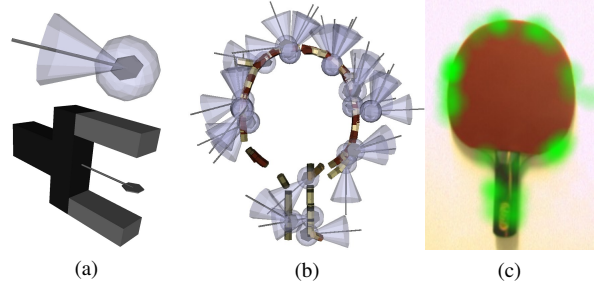


Fig. 4. Grasp density representation. The top image of Fig. (a) illustrates a particle from a nonparametric grasp density, and its associated kernel widths: the translucent sphere shows one position standard deviation, the cone shows the variance in orientation. The bottom image illustrates how the schematic rendering used in the top image relates to a physical gripper. Fig. (b) shows a 3D rendering of the kernels supporting a grasp density for a table-tennis paddle (for clarity, only 30 kernels are rendered). Fig. (c) indicates with a green mask of varying opacity the values of the location component of the same grasp density along the plane of the paddle (orientations were ignored to produce this last illustration).

The expressiveness of a single $SE(3)$ kernel (1) is rather limited: location and orientation components are both isotropic, and within a kernel, location and orientation are modeled independently. Nonparametric methods account for the simplicity of individual kernels by employing a large number of them: a grasp density will typically be supported by a thousand particles. Fig. 4a shows an intuitive rendering of an $SE(3)$ kernel from a grasp density. Fig. 4b and Fig. 4c illustrate continuous densities.

VI. LEARNING GRASP DENSITIES

This section explains how hypothesis densities are learned from source data (Section VI-A), and how empirical densities are learned from experience (Section VI-B).

A. Proposal Densities From Examples

Initial grasp knowledge, acquired for instance from imitation or reflex, is structured as a set of grasps parametrized by a 6D pose. Given the nonparametric representation, building a density from a set of grasps is straightforward – grasps can directly be used as particles representing the density. We typically limit the number of particles in a density to a thousand; if the number of grasps in a set is larger than 1000, the density is *resampled*: kernels are associated the particles, and 1000 samples are drawn and used as a representation replacement.

Since we wish to record object-relative information, densities have to be transformed to the reference frame of the object. Assuming that grasp poses are initially defined in the same reference frame as the visual ECV descriptors, this can be done by aligning the hierarchical model of the object by visual inference, and transforming the particles of each grasp density in the reference frame defined by the pose of the top feature of the aligned model.

A grasp density is integrated into the hierarchical object model through a new primitive feature i . The new feature

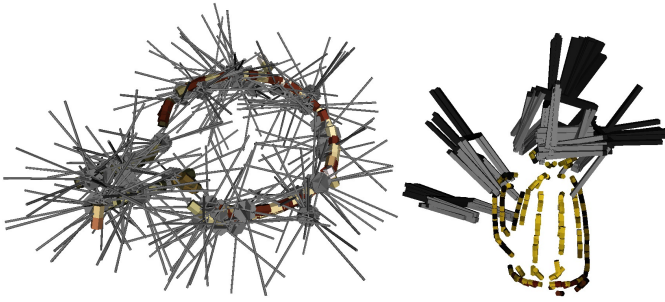


Fig. 5. Particles supporting grasp hypothesis densities.

is linked to the top model feature o through a potential $\psi_{io}(X_i, X_o)$ that corresponds to the object-relative density.

B. Empirical Densities From Familiarization

As the name suggests, hypothesis densities do not pretend to reflect the true properties of an object. Their main defect is that they may strongly suggest grasps that might not be applicable at all, for instance because of gripper discrepancies in imitation-based hypotheses. A second, more subtle issue is that the grasp data used to learn hypothesis densities will generally be afflicted with a source-dependent spatial bias. A very good example can be made from the reflex computation of Section IV-B. Reflexes are computed from ECV visual descriptors. Therefore, parts of an object that have a denser visual resolution will yield more reflexes, incidentally biasing the corresponding region of the hypothesis density to a higher value. The next paragraph explains how grasping experience can be used to compute new densities (*empirical* densities) that better reflect gripper-object properties.

Empirical densities are learned from the execution of *samples* from a hypothesis density, intuitively allowing the agent to familiarize itself with the object by discarding wrong hypotheses and refining good ones. Familiarization thus essentially consists in autonomously learning an *empirical* density from the outcomes of sample executions. A simple way to proceed is to build an empirical density directly from successful grasp samples. However, this approach would inevitably propagate the spatial bias mentioned above to empirical densities. Instead, we use importance sampling [8] to properly weight grasp outcomes, allowing us to draw samples from the physical grasp affordance of an object. The weight associated to a grasp sample x is computed as $\mathbf{a}(x) / \mathbf{q}(x)$, where $\mathbf{a}(x)$ is 1 if the execution of x has succeeded, 0 else, and $\mathbf{q}(x)$ corresponds to the value of the continuous hypothesis density at x . A set of these weighted samples directly forms a grasp empirical density that faithfully and uniformly reflects intrinsic gripper-object properties. Each empirical density is associated to the object model in the same way as proposal densities, through a new feature in the hierarchical network.

VII. RESULTS

This section illustrates hypothesis densities learned from imitation and reflexes, and empirical densities are learned by

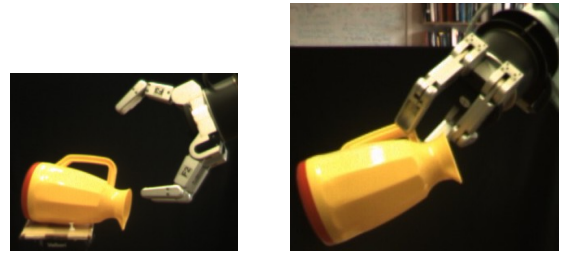


Fig. 6. Barrett hand grasping the toy jug.

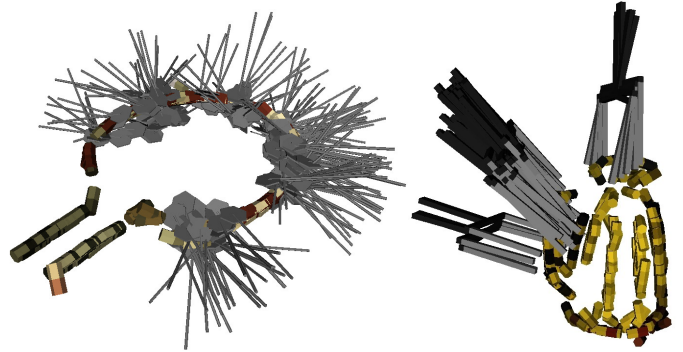


Fig. 7. Samples drawn from grasp empirical densities.

grasping objects with a 3-finger Barrett hand. Densities are built for two objects: the table-tennis paddle of Fig. 3, and a toy plastic jug (Fig. 6). The experimental scenario is described below.

For each object, the experiment starts with a visual hierarchical model, and a set of grasps. For the paddle, grasps are generated with the method described in Section IV-B. Initial data for the jug was collected through human demonstration, using a motion capture system. From these data, a hypothesis density is built for each object. The particles supporting the hypothesis densities are rendered in Fig. 5.

In order to refine affordance knowledge, feedback on the execution of hypothesis density samples is needed. Grasps are executed with a Barrett hand mounted on an industrial arm. As illustrated in Fig. 6, the hand preshape is a parallel-fingers, opposing-thumb configuration. The reference pose of the hand is set for a pinch grasp, with the tool center point located in-between the tips of the fingers – similar to the reference pose illustrated in Fig. 4a. A grasp is considered successful if the robot is able to firmly lift up the object, success being asserted by raising the robotic hand while applying a constant, inward force to the fingers, and checking whether at least one finger is not fully closed. Sets of 100 and 25 successful grasps were collected for the paddle and the jug respectively. This information was then used to build a grasp empirical density, following the procedure described in Section VI-B. Samples from the resulting empirical densities are shown in Fig. 7. For the paddle, the main evolution from hypothesis to empirical density is the removal of a large number of grasps for which the gripper wrist collides with the paddle body. Grasps presenting a steep approach relative to the plane of the paddle

were also discarded, thereby preventing fingers from colliding with the object during hand servoing. None of the pinch-grasps at the paddle handle succeeded, hence their absence from the empirical density.

While grasping the top of the jug is easy for a human hand, it proved to be very difficult for the Barrett hand with parallel fingers and opposing thumb. Consequently, a large portion of the topside grasps suggested by the hypothesis density are not represented in the empirical density. The most reliable grasps approach the handle of the jug from above; these grasps are strongly supported in the empirical density.

The left image of Fig. 7 clearly illustrates the correlation between grasp positions and orientations: moving along the edge of the paddle, grasp approaches are most often roughly perpendicular to the local edge tangent. The nonparametric density representation successfully captures this correlation.

VIII. CONCLUSION AND FUTURE WORK

We presented a framework for representing and learning object grasp affordances, and linking these to a visual object model. The affordance representation is probabilistic and nonparametric: an affordance is recorded in a continuous probability density function supported by a set of particles.

Grasp densities are initially learned from visual cues or imitation, leading to grasp hypothesis densities. Using the visual model for pose estimation, an agent is able to execute *samples* from a hypothesis density under arbitrary object poses. Observing the outcomes of these grasps allows the agent to learn from experience: an importance sampling algorithm is used to infer faithful object grasp properties from successful grasp samples. The resulting *grasp empirical densities* eventually allow for more robust grasping.

Importance Sampling is a batch learning method, that requires the execution of a large number of grasps before an empirical density can be built. Learning empirical densities *on-line* would be very convenient, which is a path we plan to explore next.

We currently learn visual and grasp features independently, and connect them through a single top-level model feature. Yet, a part-based representation offer an elegant way to *locally* encode visuomotor descriptions. One of our goals is to learn visual parts that share the same grasp properties across different objects. This way, a grasp feature will be directly and exclusively connected to the visual evidence that predicts its applicability, allowing for its generalization across objects.

ACKNOWLEDGMENTS

This work was supported by the Belgian National Fund for Scientific Research (FNRS) and the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657). We thank Volker Krüger and Dennis Herzog for their support during the recording of the human demonstration data.

REFERENCES

- [1] Daniel Aarno, Johan Sommerfeld, Danica Kragic, Nicolas Pugeault, Sinan Kalkan, Florentin Wörgötter, Dirk Kraft, and Norbert Krüger. Early reactive grasping with second order 3D feature relations. In *The IEEE International Conference on Advanced Robotics*, 2007.
- [2] G. Biegelbauer and M. Vincze. Efficient 3D object detection by fitting superquadrics to range image data for robot's object manipulation. In *IEEE International Conference on Robotics and Automation*, 2007.
- [3] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *Computer Vision and Pattern Recognition*, volume 1, pages 710–715, 2005.
- [4] C. de Granville and A. H. Fagg. Learning grasp affordances through human demonstration. *submitted to the Journal of Autonomous Robots*, 2009.
- [5] Charles de Granville, Joshua Southerland, and Andrew H. Fagg. Learning grasp affordances through human demonstration. In *Proceedings of the International Conference on Development and Learning (ICDL'06)*, 2006.
- [6] Renaud Detry and Justus H. Piater. Hierarchical integration of local 3D features for probabilistic pose recovery. In *Robot Manipulation: Sensing and Adapting to the Real World (Workshop at Robotics, Science and Systems)*, 2007.
- [7] Renaud Detry, Nicolas Pugeault, and Justus H. Piater. Probabilistic pose recovery using learned hierarchical object models. In *International Cognitive Vision Workshop (Workshop at the 6th International Conference on Vision Systems)*, 2008.
- [8] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
- [9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient matching of pictorial structures. In *Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, pages 2066–, 2000.
- [10] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [11] Kai Huebner, Steffen Ruthotto, and Danica Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. Technical report, KTH, 2007.
- [12] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragic, S. Kalkan, F. Wörgötter, and N. Krüger. Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes. *Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics*, 2008. (accepted).
- [13] Danica Kragic, Andrew T. Miller, and Peter K. Allen. Real-time tracking meets online grasp planning. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 2460–2465, 2001.
- [14] N. Krüger, M. Lappe, and F. Wörgötter. Biologically Motivated Multimodal Processing of Visual Primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428, 2004.
- [15] James Kuffner. Effective sampling and distance metrics for 3D rigid body path planning. In *Proc. 2004 IEEE Int'l Conf. on Robotics and Automation (ICRA 2004)*. IEEE, May 2004.
- [16] K. V. Mardia and P. E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 1999.
- [17] A. T. Miller, S. Knoop, H. Christensen, and P. K. Allen. Automatic grasp planning using shape primitives. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2003*, volume 2, pages 1824–1829, 2003.
- [18] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [19] Nicolas Pugeault. *Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation*. Vdm Verlag Dr. Müller, 2008.
- [20] Mario Richtsfeld and Markus Vincze. Robotic grasping based on laser range and stereo data. In *International Conference on Robotics and Automation*, 2009.
- [21] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [22] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

Probabilistic Pose Recovery Using Learned Hierarchical Object Models

Renaud Detry¹, Nicolas Pugeault², and Justus Piater¹

¹ Université de Liège, Liège, Belgium,

Renaud.Detry@ULg.ac.be, Justus.Piater@ULg.ac.be

² University of Southern Denmark, Odense, Denmark,
The University of Edinburgh, Edinburgh, Scotland, UK,
npugeaul@inf.ed.ac.uk

Abstract. This paper presents a probabilistic representation for 3D objects, and details the mechanism of inferring the pose of real-world objects from vision. Our object model has the form of a hierarchy of increasingly expressive 3D features, and probabilistically represents 3D relations between these. Features at the bottom of the hierarchy are bound to local perceptions; while we currently only use visual features, our method can in principle incorporate features from diverse modalities within a coherent framework. Model instances are detected using a Nonparametric Belief Propagation algorithm which propagates evidence through the hierarchy to infer globally consistent poses for every feature of the model. Belief updates are managed by an importance-sampling mechanism that is critical for efficient and precise propagation. We conclude with a series of pose estimation experiments on real objects, along with quantitative performance evaluation.

Keywords Computer vision, 3D object representation, pose estimation, Non-parametric Belief Propagation.

1 Introduction

The merits of part-based and hierarchical approaches to object modelling have often been put forward in the vision community [9,5,11]. Part-based representations are more robust to occlusions and viewpoint changes than global representations, and spatial configurations increase their expressiveness. Moreover, they not only allow for bottom-up inference of object parameters based on features detected in images, but also for top-down inference of image-space appearance based on object parameters.

The advantages of visual part-based representations naturally extend to multi-sensory cases. For example, haptic and proprioceptive information won't relate to an object as a whole. Instead, they typically emerge from specific grasps, on specific parts of the object. Part-based representation offer a neat way to *locally* encode cross-modal descriptions that emphasise the relations between the different types of percepts.

We are currently developing a 3D, part-based object representation framework, along with mechanisms for unsupervised learning and probabilistic inference of the model. Our model combines local appearance and 3D spatial relationships through a hierarchy of increasingly expressive *features*. Features at the bottom of the hierarchy are bound to local visual perceptions. Features at other levels represent combinations of more elementary features, and encode probabilistic relative spatial relationships between their children. The top level of the hierarchy contains a single feature which represents the object.

To detect instances of a model in a cluttered scene, evidence is propagated throughout the hierarchy by probabilistic inference mechanisms, leading to one or more consistent scene interpretations: the model is able to suggest a number of likely *poses* for the object, a pose being composed of a 3D location and a 3D body orientation defined in the reference frame of the camera that captured the raw visual data. The use of probabilistic inference algorithms permits the uniform integration of all available evidence, allowing for unbiased contributions of all low-level features.

In previous work [2], we presented a learning method that constructs a hierarchy from a set of object observations. We also gave an overview of an inference process that followed a straightforward Nonparametric Belief Propagation scheme [12] and allowed for pose recovery of artificial objects. In this paper, we present in greater detail a significantly improved version of this inference process. We added an importance-sampling (IS) message product suggested in a similar form by Ihler et al. [6], and extended it to a two-level IS sampling of *implicit* message products which is readily applicable to pose estimation on real-world objects.

Unsupervised learning, probabilistic representation and robust detection are three aspects that we believe make our representation a good candidate for the perception and memory tasks of a cognitive system. Furthermore, the features organized in the hierarchies are not especially restricted to one input modality. We currently work with visual input only, but our model is intended to unite different types of perceptual information, e.g. vision plus haptic and proprioceptive inputs simultaneously. This will produce cross-modal descriptions and cross-modal behaviors directly applicable to action-related tasks such as grasping and object manipulation, as a grasp strategy may be linked directly to visual features that predict its applicability.

We emphasize that we are not developing an object classification framework. Object classification is best achieved using *discriminative* models and presupposes the presence of one object to be classified. Instead, we intend to develop *object-centric* representations that allow for detection and localisation of known objects within a highly cluttered scene. Also, our representations lend themselves to applications other than classification (e.g. manipulation).

2 Hierarchical Model

Our object model consists of a set of generic *features* organized in a hierarchy. Features that form the bottom level of the hierarchy, referred to as *primitive features*, are bound to visual observations. The rest of the features are *meta-features* which embody spatial configurations of more elementary features, either meta or primitive. Thus, a meta-feature incarnates the relative configuration of two features from a lower level of the hierarchy.

A feature can intuitively be associated to a “part” of an object, i.e. a generic component instantiated once or several times during a “mental reconstruction” of the object. At the bottom of the hierarchy, primitive features correspond to local parts that each may have many *instances* in the object. Climbing up the hierarchy, meta-features correspond to increasingly complex parts defined in terms of constellations of lower parts. Eventually, parts become complex enough to satisfactorily represent the whole object.

Formally, the hierarchy is implemented in a Pairwise Markov Random Field. Features correspond to hidden nodes of the network. When a model is associated to a particular scene (during construction or instantiation), the pose of feature i in that scene will be represented by the probability density function of the random variable x_i associated to feature i , effectively linking feature i to its instances. Random variables are thus defined over the pose space $SE(3) = \mathbb{R}^3 \times SO(3)$.

The structure of the hierarchy is reflected by the edge pattern of the network; each meta-feature is thus linked to its two child features. As noted above, a meta-feature encodes the relationship between its two children. However, the graph records this information in a slightly different but equivalent way: instead of recording the relationship between the two child features, the graph records the two relationships between the meta-feature and each of its children. The relationship between a meta-feature i and one of its children j is parametrized by a *compatibility potential function* $\psi_{ij}(x_i, x_j)$ associated to the edge e_{ij} . A compatibility potential specifies, for any given pair of poses of the features it links, the probability of finding that particular configuration for these two features. We only consider rigid-body relationships. Moreover, relationships are *relative* spatial configurations. Compatibility potentials can thus be represented by a probability density over the feature-to-feature transformation space $SE(3)$.

Finally, each primitive feature is linked to an observed variable y_i . Observed variables are tagged with an appearance descriptor called a *codebook vector*. The set of all codebook vectors forms a *codebook* that binds the object model to feature observations. The statistical dependency between a hidden variable x_i and its observed variable y_i is parametrized by an *observation potential* $\phi_i(x_i)$, also referred to as *evidence* for x_i , which corresponds to the spatial distribution of the observations. We generally cannot observe meta-features; their observation potential is thus uniform.

3 Inference

Model instantiation is the process of detecting instances of an object model in a scene. It provides pose densities for all features of the model, indicating where the learned object is likely to be present. Instantiating a model in a scene amounts to inferring posterior marginal densities for all features of the hierarchy.

The first step of inference is to define priors (observation potentials, evidence) for all features (hidden nodes) of the model. For primitive features, evidence is estimated from feature observations. Observations are classified according to the observation codebook; for each primitive feature i , its observation potential $\phi_i(x_i)$ is estimated from observations that are (softly) associated to the i^{th} codebook vector. For meta-features, evidence is uniform.

Once priors have been defined, instantiation can be achieved by any applicable inference algorithms. We currently use a Belief Propagation algorithm of which we give a complete, top-down view below.

3.1 Belief Propagation

Belief Propagation (BP) [10,13,7] is based on incremental updates of marginal probability estimates, referred to as *beliefs*. The belief at feature i is denoted by

$$b(x_i) \approx \mathbf{P}(x_i|y) = \int \dots \int \mathbf{P}(x_1, \dots, x_N|y) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_N$$

where y stands for the set of observations. During the execution of the algorithm, *messages* are exchanged between neighboring features (hidden nodes). A message that feature i sends to feature j is denoted by $m_{ij}(x_j)$, and contains feature i 's belief about the state of feature j . In other words, $m_{ij}(x_j)$ is a real positive function proportional to feature i 's belief about the plausibility of finding feature j in pose x_j . Messages are exchanged until all beliefs converge, i.e. until all messages that a node receives predict a similar state.

At any time during the execution of the algorithm, the current pose belief (or marginal probability estimate) for feature i is the normalized product of the local evidence and all incoming messages, as

$$b_i(x_i) = \frac{1}{Z} \phi_i(x_i) \prod_{j \in \text{neighbors}(i)} m_{ji}(x_i), \quad (1)$$

where Z is a normalizing constant. To prepare a message for feature j , feature i starts by computing a "local pose belief estimate", as the product of the local evidence and all incoming messages *but* the one that comes from j . This product is then multiplied with the compatibility potential of i and j , and marginalized over x_i . The complete message expression is

$$m_{ij}(x_j) = \int \psi_{ij}(x_i, x_j) \phi_i(x_i) \prod_{k \in \text{neighbors}(i) \setminus j} m_{ki}(x_i) dx_i. \quad (2)$$

As we see, the computation of a message doesn't directly involve the complete local belief (1). In general, the explicit belief for each node is computed only once, after all desirable messages have been exchanged.

When BP is finished, collected evidence has been propagated from primitive features to the top of the hierarchy, permitting inference of the top feature marginal pose density. Furthermore, regardless of the propagation scheme (message update order), the iterative aspect of the message passing algorithm ensures that global belief about the object pose – concentrated at the top nodes – has at some point been propagated back down the hierarchy, reinforcing globally consistent evidence and permitting the inference of occluded features. While there is no theoretical proof of BP convergence for loopy graphs, empirical success has been demonstrated in many situations.

3.2 Nonparametric Representation

We opted for a nonparametric approach to probability density representation for all entities of the model, i.e. random variable and functions of random variables, including potentials, messages, and evidence. A density is simply represented by a set of (possibly weighted) particles; the local density of these particles in a given region is proportional to the actual probabilistic density in that region. The number of particles supporting a density is fixed, and will be denoted by n . Whenever a density has to be evaluated, traditional kernel density estimation methods can be used. Compared to usual parametric approaches that involve a limited number of parametrized kernels, a nonparametric approach eliminates problems like fitting of mixtures or the choice of a number of components. Also, no assumption concerning the shape of the density has to be made.

Figure 1 shows an example of a hierarchy for a traffic sign. Feature 2 is a primitive feature that corresponds to a local black-white edge segment – the white looks greenish on the picture. The blue patch pattern in the $\phi_2(x_2)$ box is the non-parametric representation for the evidence distribution for feature 2. The blue patch pattern in the x_2 box is the non-parametric representation for the posterior density of x_2 , i.e. the poses in which part “feature 2” is likely to be found. Feature 4 is the combination of primitive features 1 and 2. The red patch in the x_4 box shows its inferred pose in the scene. The $\psi_{4,2}(x_4, x_2)$ box shows the encoding of the relationship between features 4 and 2; for a fixed pose for feature 4 (in red), it shows the likely poses for feature 2 (in blue). The sign itself corresponds to feature 6, denoted by its random variable x_6 . It is the composition of two features, one representing the central “opening bridge” pattern *and* the corners of the inner triangle (feature 4), the other representing the central pattern *and* the outer edges (feature 5).

3.3 Nonparametric Belief Propagation

For inference, we use a variant of BP, Nonparametric Belief Propagation (NBP), an algorithm for BP message update in the particular case of continuous, non-Gaussian potentials [12]. The underlying method is an extension of particle fil-

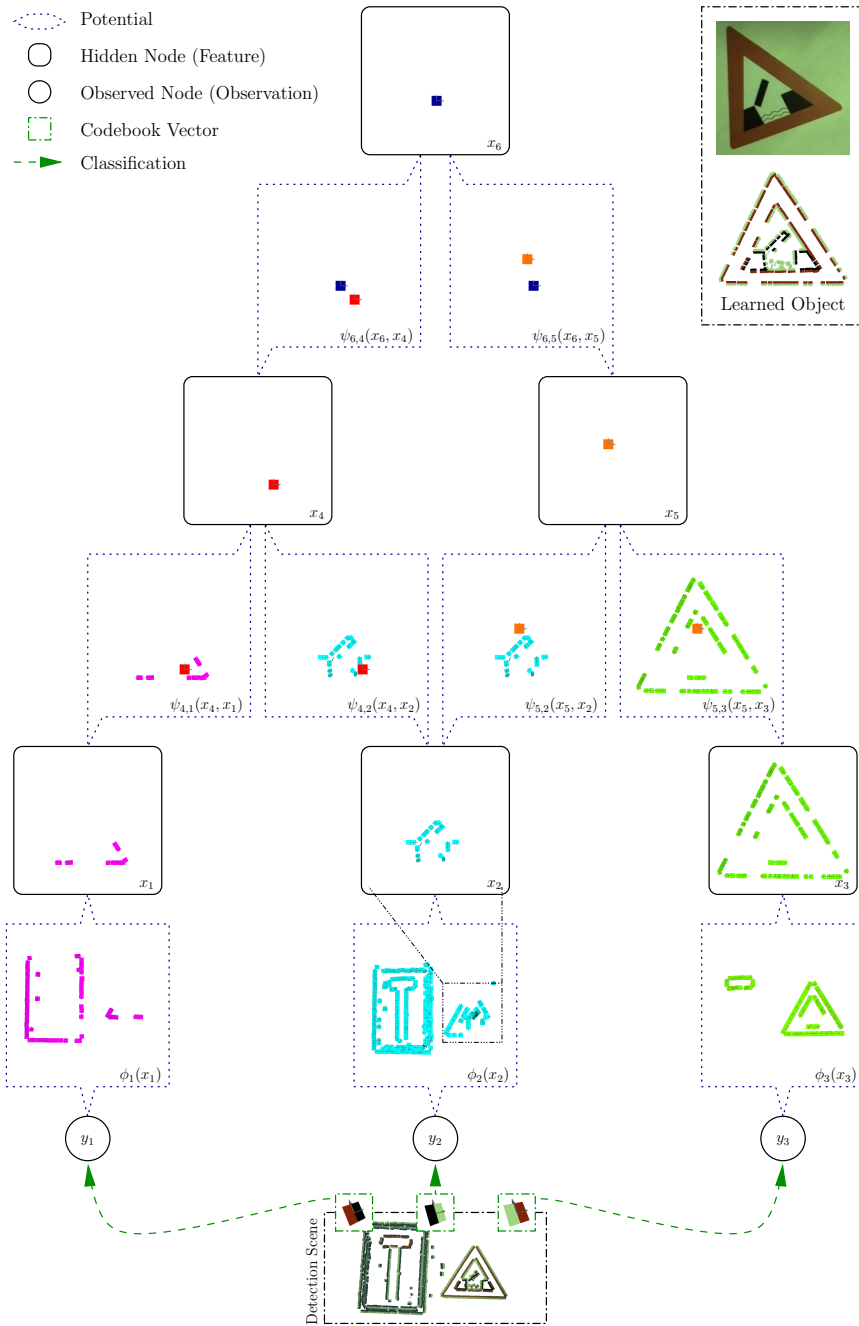


Fig. 1. Example of a hierarchical model of a traffic sign.

tering; the representational approach is thus nonparametric and fits our model very well.

NBP is easier to explain if we decompose the analytical message expression (2) into two steps:

1. Computation of the local belief estimate

$$\beta_{ts}(x_t) = \phi_t(x_t) \prod_{i \in N(t) \setminus s} m_{it}(x_t), \quad (3)$$

2. Combination of β_{ts} with the compatibility function ψ_{ts} , and marginalisation over x_t

$$m_{ts}(x_s) = \int \psi_{ts}(x_t, x_s) \beta_{ts}(x_t) dx_t. \quad (4)$$

NBP forms a message by first sampling from the product (3) to collect a non-parametric representation for $\beta_{ts}(x_t)$, it then samples from the integral (4) to collect a non-parametric representation for $m_{ts}(x_s)$. These two operations are executed alternately: transform local estimate to form a message, merge messages to form a local estimate, etc...

Sampling from the message product (3) is conceptually straightforward. Using Gaussian kernel density estimation, each factor (messages and evidence) can be represented by a weighted sum of n Gaussians. The product of a series of Gaussians is also a Gaussian, and the parameters (mean, variance, weight) of the product Gaussian can easily be computed from the parameters of the factor Gaussians. Hence, letting $d = (N(t) - 1) + 1$ denote the number of factors in the product (3), $\beta_{ts}(x_t)$ can be expressed as a weighted sum of n^d Gaussians [12]. A nonparametric representation for $\beta_{ts}(x_t)$ can thus be constructed by sampling from a mixture of n^d Gaussians, which amounts to repetitively selecting one Gaussian at random and taking a random sample from it. The computational cost of this exhaustive approach is $O(n^d)$. Clearly, exhaustive product implementations will suffer from overly long computation times.

The second phase of the NBP message construction computes an approximation for the integral (4) by stochastic integration. Stochastic integration takes a series of samples $\hat{x}_t^{(i)}$ from $\beta_{ts}(x_t)$, and propagates them to feature s by sampling from $\psi_{ts}(\hat{x}_t^{(i)}, x_s)$ for each $\hat{x}_t^{(i)}$. It would normally also be necessary to take into account the marginal influence of $\psi_{ts}(x_t, x_s)$ on x_t . In our case however, potentials only depend on the difference between their arguments; the marginal influence is a constant and can be ignored.

3.4 Importance Sampling

The computational bottleneck of NBP clearly lies in message products. Ihler et al. explored multiple improvements over the exhaustive product [6], one of which is to sample from the product using Importance Sampling (IS). IS is a technique for sampling from an unknown distribution $p(x)$ by sampling a series of examples $\hat{x}^{(\ell)}$ from a known distribution $q(x)$ ideally similar to p . IS accounts

for the difference between the target distribution p and the proposal distribution q by assigning to each sample a weight defined as

$$w^{(\ell)} = \frac{p(\hat{x}^{(\ell)})}{q(\hat{x}^{(\ell)})}.$$

To produce a sample of size n , one usually takes rn weighted examples from q , where $r > 1$, and eventually resamples them to a size of n . The closer q is to p , the better $\{\hat{x}^{(\ell)}\}$ will approximate p .

Sampling from a message product (3) with IS works by selecting one of the messages $m_{ut}(x_t)$ (or the evidence) as proposal distribution, the rest of the factors providing importance weights:

$$w^{(\ell)} = \frac{\phi_t(\hat{x}_t^{(\ell)}) \prod_{i \in N(t) \setminus s} m_{it}(\hat{x}_t^{(\ell)})}{m_{ut}(\hat{x}_t^{(\ell)})} = \phi_t(\hat{x}_t^{(\ell)}) \prod_{i \in N(t) \setminus \{s, u\}} m_{it}(\hat{x}_t^{(\ell)}).$$

IS produces n samples from a product of d factors in $O(rdn^2)$ time. From here on, we will consider that the number of neighbors a node may have is bounded and typically low, and ignore it in complexity statements. IS thus produces n samples from a product of d factors in $O(rn^2)$ time.

4 Efficient Importance Sampling of Message Products

The success of NBP inference highly depends on a sufficient density resolution, i.e. having enough particles to support the different modes of potentials, local estimates, and messages. Moving to more complex applications will generally require an increase of n , which has a hard impact on computational time and memory needs. This section presents a variant of the IS-based NBP algorithm that yields a significant improvement of the inference power without any memory impact. Its computational behavior is close to original IS-based NBP, with some interesting benefits.

4.1 Representational Constraints

As explained above, A message that feature i sends to feature j – denoted by $m_{ij}(x_j)$ – contains feature i 's belief about the state of feature j . Feature i will often possess a rather inaccurate local estimate, e.g. at the beginning of propagation when each bottom feature receives observations from the whole scene surrounding an object of interest. Additionally, even if a local estimate was exact, transforming it with ψ_{ij} will generate a large number of possible states for feature j , only a fraction of which will eventually become confirmed by other messages incoming to j – the job of message products precisely is to extract sections that overlap between incoming messages. Generating a message from local estimates can be pictured as an exploration process, while merging messages together would be a confirmation/concentration process. From these

observations, it intuitively follows that one may achieve better performance by increasing the resolution of messages only, leaving potentials and local estimates at their initial resolution.

4.2 Implicit Messages

Let us now turn to the propagation equation (2), which we *analytically* decomposed into a multiplication (3) and an integration (4). We explained that NBP implements BP by *physically* performing the same decomposition, i.e. computing explicit nonparametric representations for messages and local estimates alternately. In this section, we propose a somewhat different implementation, in which explicit representations are only computed for local estimates.

Let us assume we are in the process of constructing a nonparametric representation for $\beta_{ts}(x_t)$, i.e. the local estimate of feature t that includes all incoming information but that from s . In typical IS-based NBP, we first choose one incoming message $m_{ut}(x_t)$ at random ($u \neq s$) as IS proposal density; then, we repetitively take a sample $\hat{x}_t^{(\ell)}$ from $m_{ut}(x_t)$ and compute its importance weight

$$w^{(\ell)} = \phi_t(\hat{x}_t^{(\ell)}) \prod_{i \in N(t) \setminus \{s, u\}} m_{it}(\hat{x}_t^{(\ell)}). \quad (5)$$

One can notice though that neither of these two operations do actually need an explicit expression for incoming messages. Producing $\hat{x}_t^{(\ell)}$ from $\beta_{ut}(x_t)$ and $\psi_{ut}(x_u, x_t)$ is straightforward. In turn, Expression (5) can be rewritten

$$w^{(\ell)} = \phi_t(\hat{x}_t^{(\ell)}) \prod_{i \in N(t) \setminus \{s, u\}} \int \psi_{it}(x_i, \hat{x}_t^{(\ell)}) \beta_{it}(x_i) dx_i. \quad (6)$$

Evaluating each integral is achieved by sampling p times an example $\hat{x}_i^{(k)}$ from either $\psi_{it}(x_i, \hat{x}_t^{(\ell)})$ or $\beta_{it}(x_i)$, evaluating $\beta_{it}(\hat{x}_i^{(k)})$ or $\psi_{it}(\hat{x}_i^{(k)}, \hat{x}_t^{(\ell)})$ respectively, and taking the average over k .

The computational complexity of importance weight computation with explicit messages (5) is $O(n)$, because of linear iteration through all messages and evidence which are of size n . The computational complexity with implicit messages (6) is $O(pn)$, because of p linear iterations through potentials or the local estimates. However, implicit messages effectively achieve the same resolution as explicit messages would *if* these explicit messages were supported by pn particles, *while keeping memory needs at $O(n)$* . Importance weight computation with implicit or explicit messages are thus expected to display processing times of the same order, while the implicit method will categorically require less memory.

4.3 Two-Level Importance Sampling

One known weakness of IS-based NBP is that it cannot intrinsically concentrate its attention on the modes of a product, which is an issue since individual messages often present many irrelevant modes [6]. We overcome this problem with

a two-level IS: we first compute an intermediate representation for the product with the procedure explained above, we then use this very representation as the proposal distribution for a second IS that will be geared towards relevant modes. The intermediate representation is obtained with sparse implicit messages ($p \ll n$) but many importance samples ($r \gg 1$), while the second IS uses rich implicit messages ($p \approx n$) but a low value for r . Denoting by $\beta_{ts}^*(x_t)$ the intermediate product representation, importance weights for the second IS are computed as

$$w^{(\ell)} = \frac{\phi_t(\hat{x}_t^{(\ell)}) \prod_{i \in N(t) \setminus s} m_{it}(\hat{x}_t^{(\ell)})}{\beta_{ts}^*(\hat{x}_t^{(\ell)})}.$$

In the equation above, messages are implicit.

The two-level IS described above and the high-resolution messages have been crucial elements of the successful application to real-world object presented at Section 5.2.

5 Evaluation

5.1 Pose Estimation

The feature at the top of a hierarchical object model represents the whole object. When instantiating the model in a scene in which exactly one instance of the object is present, the top feature density should present one major mode, which can be used to estimate the object pose. Let us consider a model for a given object, and a pair of scenes where the object appears. In the first scene, the object is in a reference pose. In the second scene, the pose of the object is unknown. The application of our method to estimate the pose of the object in the second scene goes as follows:

1. Instantiate the object model in the reference scene, and compute a *reference object pose* π_1 as the mean of the top feature density major mode.
We emphasize that a hierarchy comes from *unsupervised* recursive combinations of features [2]. Even though the *object* is in a reference pose, π_1 is not expected to be located at $(0, 0, 0)$ or aligned with $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, which makes this first step necessary.
2. Instantiate the object model in the unknown scene and compute pose π_2 from the major mode of the top feature density.
3. Let t be the transformation between π_1 and π_2 . This transformation corresponds to the rigid body motion between the pose of the object in the first scene and its pose in the second scene. Since the first scene is a reference pose, t is the *pose* of the object in the second scene.

A prominent aspect of this procedure is its ability to recover an object pose without explicit point-to-point correspondences. The estimated pose emerges from a negotiation involving all available data.

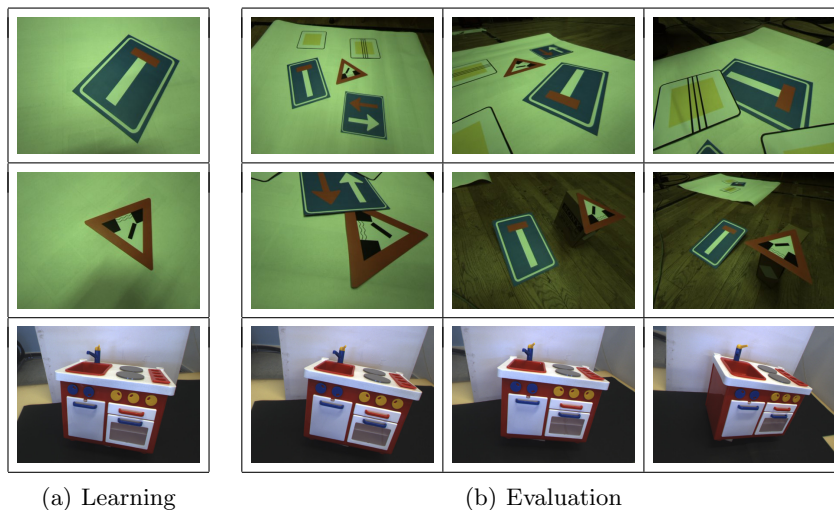


Fig. 2. Input imagery (only the left image in each stereo pair is presented). Effective resolution is 1280×960 pixels.

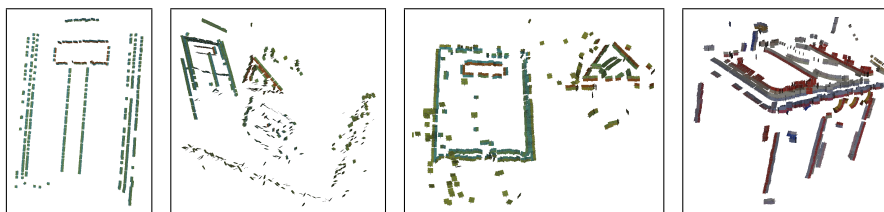


Fig. 3. Examples of ECV representations, extracted from scenes of Figure 2.

5.2 Experiments

In this section, we demonstrate the applicability of our model with a series of pose estimation experiments in various cluttered scenes. We chose to learn models for the three objects presented at Figure 2(a). We then tried to estimate their poses in the scenes of Figure 2(b).

Observations are provided by an early-cognitive-vision (ECV) system [8], which extracts 3D primitives from stereo views of a scene. The quality of such ECV representations varies as a function of local visual signal quality. Figure 3 illustrates the ECV primitives for certain scenes of Figure 2.

Models for the three objects of Figure 2(a) were learned following the procedure mentioned above [2]. These models were learned from a clean view of each object (the reference scene), for example from the ECV representation in the first image of Figure 3. Each model has also been instantiated in its reference scene to compute its reference pose π_1 .

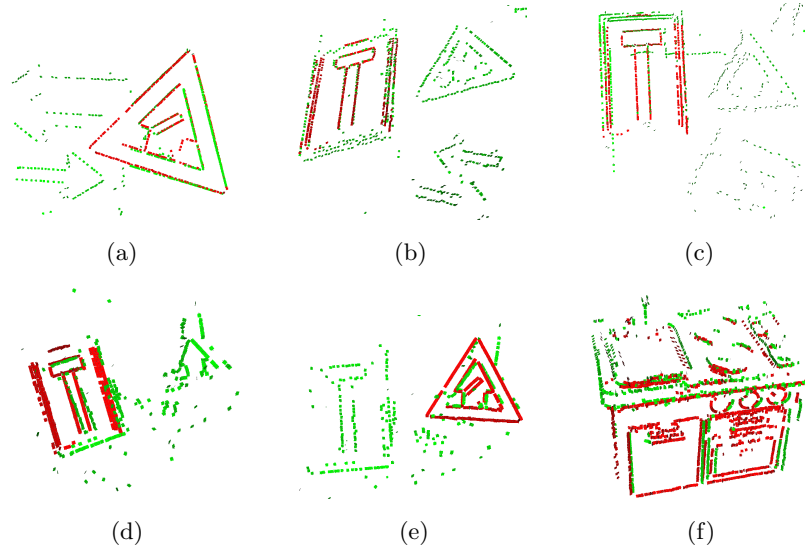


Fig. 4. Illustration of the pose estimation accuracy. Each picture shows in green a scene that contains one object of interest and in red the pose of that object inferred by our system.

The three models were all instantiated in the test scenes of Figure 2(b), using observations like these of Figure 3 as evidence. Looking closer at the instantiation of one model in one scene, there are two cases to consider. First, the model had no instance in the scene. The top-feature density was then relatively uniform, and the experiment did not need to go any further. In the second case, an instance was present. It was then always verified that the top feature did present a principal mode π_2 . We could thus compute the transformation t between π_1 and π_2 , which corresponds to the *estimated* rigid body motion between the pose of the object in the reference scene, and to its pose in the noisy scene.

We can evaluate the success of the experiment by transforming the reference scene with t , and superimposing it onto the test scene; if the experiment is successful, the object of interest should overlap with its instance. Such evaluations are presented at Figure 4. All the experiments that we ran ended with successful pose recovery. For traffic signs, the worst estimate (Figure 4(d)) corresponds to the dead-end signal pose estimation in the sixth scene of Figure 2(b) (second row, third column). This is however one of the most difficult scenes: it has a brown background, thus changing the outside color of ECV primitives on the traffic sign contours. This induces wrong associations of observations to primitive features, and makes for harder inference. Estimation is still quite accurate given the difficulty of the scene. Other typical estimates are presented at Figure 4. In particular, 4(a) shows a good result despite occlusion.

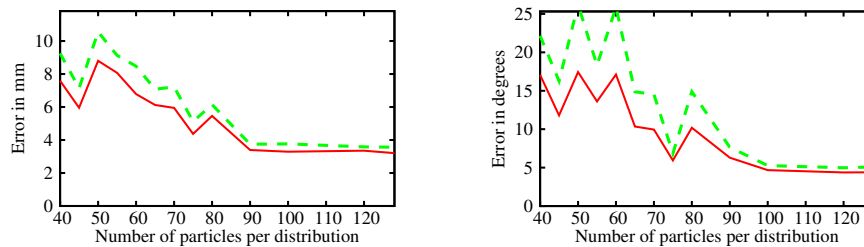


Fig. 5. Pose estimation accuracy as a function of the number of particles per density, for an instantiation of the opening-bridge traffic sign within the first scene of Figure 2(b). Left and right plots correspond to location and orientation error respectively. The red lines indicate the mean absolute error. The green lines indicate the variance across runs. Location error can be compared to the traffic sign edge, which is 190mm long. See the text for details.

The accuracy of probabilistic pose estimation highly depends on the resolution of the representation. When an experiment lacks accuracy, retrying with more particles usually produces better results. Therefore, a meaningful quantitative evaluation must take into account the number of particles per density. Figure 5 shows the pose estimation error as a function of the number of particles per density. Because of the probabilistic nature of inference, runs with different software random seeds produce different results. Therefore, we run each experiment several times and study the mean error, plotted in red in the figure. The mean error decreases quickly when going from 40 to 100 particles, and stabilizes for higher resolutions. We also plotted one standard deviation above the mean error, in dashed green. The error variance also decreases as the number of particles increases.

6 Discussion

6.1 Related Work

Compared to recent work in the field [4,3,1], the most distinguishing aspects of our approach are its explicit 3D support and the unbiased contributions of all low-level features. We learn from observations defined in 3D, and infer a full 3D pose. The use of a sophisticated inference algorithm permits the uniform integration of all available evidence, avoiding an explicit combinatorial search.

6.2 Conclusion

We presented an object representation framework that encodes probabilistic relations between 3D features. We discussed an Importance-Sampling-NBP inference process which, together with the learning scheme of our previous work [2], allow us to learn unsupervised part representations for real objects and to

instantiate them in cluttered scenes. We are thus able to achieve pose recovery without prior object models, and without explicit point correspondences.

Our method can in principle incorporate features from more perceptual modalities than vision. Our objective is to observe haptic and kinematic features that correlate with successful grasps, and integrate them into the feature hierarchy. Then, given a visual scene, grasp parameters can be suggested by probabilistic inference within the feature hierarchy.

Acknowledgment

This work was supported by the Belgian National Fund for Scientific Research (FNRS) and the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657).

References

1. G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *Computer Vision and Pattern Recognition*, volume 1, pages 710–715, 2005.
2. Renaud Detry and Justus H. Piater. Hierarchical integration of local 3D features for probabilistic pose recovery. In *Robot Manipulation: Sensing and Adapting to the Real World (Workshop at Robotics, Science and Systems)*, 2007.
3. Boris Epshtein and Shimon Ullman. Feature hierarchies for object classification. In *IEEE International Conference on Computer Vision*, 2005.
4. Sanja Fidler and Aleš Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR07*, 2007.
5. Kuniyiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.
6. Alexander T. Ihler, Erik B. Sudderth, William T. Freeman, and Alan S. Willsky. Efficient multiscale sampling from products of Gaussian mixtures. In *Neural Information Processing Systems*, 2003.
7. Michael I. Jordan and Yair Weiss. Graphical models: Probabilistic inference. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, 2nd edition. MIT Press, 2002.
8. Norbert Krüger and Florentin Wörgötter. Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. In Massimo De Gregorio, Vito Di Maio, Maria Frucci, and Carlo Musio, editors, *BVAI*, volume 3704 of *Lecture Notes in Computer Science*, pages 157–166. Springer, 2005.
9. David Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three dimensional shapes. *Proceedings of the Royal Society of London B*, 200:269–294, 1978.
10. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
11. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 1999.
12. Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. *cvpr*, 01:605, 2003.
13. Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2002.

**GROUNDING ATTENTION IN ACTION CONTROL: THE INTENTIONAL
CONTROL OF SELECTION**

Bernhard Hommel

Leiden University
Institute for Psychological Research & Leiden Institute for Brain and Cognition
Leiden, The Netherlands

Corresponding Address:
Bernhard Hommel
Leiden University
Department of Psychology
Cognitive Psychology Unit
Postbus 9555
2300 RB Leiden, The Netherlands
hommel@fsw.leidenuniv.nl

ABSTRACT

The main function of human attention is commonly thought to consist in preventing information overload of the cognitive system. In contrast, this chapter provides empirical evidence and theoretical reasons to consider attention a mere derivative of action control. It argues that the existence of distributed representations and concurrent processing streams creates specific control problems. Parts of these problems, so goes the claim, are solved by associating categories of actions (such as reaching or grasping) with particular perceptual dimensions in such a way that planning an action biases the cognitive system towards feature dimensions that are suited to specify the action's open parameters. This approach has major implications for attentional theory in general and the issue of effortless attention in particular.

My first poster presentation at a scientific meeting was no success. I offered a new theoretical framework on stimulus and response representation (the later Theory of Event Coding [Hommel, Müsseler, Aschersleben, & Prinz, 2001a]) together with supportive data and hoped to attract the interest of all the big shots working on stimulus-response compatibility. But no one came. One year later I presented a much less inspired study but made one crucial move: I put the A-word in the title, with the effect that my poster was one of the most crowded, and long after the session was over I was still heavily engaged in discussions. This is just one of many examples demonstrating that cognitive scientists love attention as a topic. In contrast to sensory and motor processes, say, which rather smell like hardware and mechanics, the concept of attention seems to directly connect to what makes us human, as it somehow expresses our individual needs and wishes, preferences and interests. The drawback of this attractiveness is that the concept is more often than not used as a wastebasket, a container that serves as a pseudo-explanation for the phenomena we still fail to understand—so that “attention” is explained by the workings of an “attentional system.”

One of the more successful strategies to tackle this problem is to focus on the function of attentional processes, that is, to ask what attention does rather than what it is. Indeed, the modern cognitive sciences have benefited greatly from this strategy, even though over the years we have seen rather dramatic changes in the way the functions of attention have been characterized. In the following, I will briefly discuss some of the more influential perspectives, which all have their benefits and their drawbacks. This discussion (for broader treatments, see Allport, 1993; Neumann, 1987; and Schneider, 1995) will reveal that early approaches emphasized attentional function subserving higher order cognition and consciousness, whereas more recent approaches increasingly appreciate the importance of attentional processes for action (selection-for-action). In this chapter, I would like to push this trend one step further by arguing that attention does not only subserve action-control

processes but may actually have emerged to solve action-control problems in a cognitive system that relies on distributed representations and multiple, loosely connected processing streams.

THE FUNCTIONS OF ATTENTION

Most of the grand, influential attentional theories have considered attention as a mechanism that administers and organizes scarcity. In the 19th century authors were mainly impressed by the limits of consciousness, which was assumed to be restricted to the representation of only one thought or event at a time (e.g., James, 1890). Given the emphasis on introspective methods, this limitation was rarely systematically investigated but taken for granted, and attention was thought to make the best of it. The main idea was that if consciousness can only contain one event, then attention better ensures that this event is of optimal use, which can be guaranteed by directing attention to relevant events (the endogenous aspect of attention) and having attention attracted by interesting events (the exogenous aspect of attention).

Even though modern cognitive approaches more or less did away with introspective methods, the assumed function of attention did not change much. In view of the increasing importance and availability of computers, researchers like Broadbent (1958) replaced consciousness by working memory as the central processing unit, which, however, was considered to be equally limited in processing capacity. Accordingly, attentional mechanisms were thought of as filters that discriminate between relevant and irrelevant information, and effectively gate out the latter in order to prevent working memory from overload. Again, the filters were thought to be endogenously controlled in principle, but this control could be overruled by overlearned or highly important stimuli. Emphasis on the coordinative and administrative aspects of attention was replaced by capacity theories (e.g., Kahneman, 1973), which considered the flexible use of attentional resource policies and selection strategies in

multiple task performance and everyday life. But the main function of attention was again to prevent a central processing unit from overload by gating out irrelevant information.

Recent attentional theories are more broadly informed by neuroscientific knowledge about the structure and processing characteristics of the primate brain and thus necessarily more complex. Some theories are particularly interested in the spatial limitations of attention or, more precisely, in the apparent limitation of the brain to integrate information from only one point in space at one time (e.g., Treisman, 1988; Wolfe, 1994). Other approaches are less pessimistic with regard to strict spatial limitations, but they do assume that attended locations are processed at a higher spatial resolution (e.g., Bundesen, Habekost, & Killingsbaek, 2005). Even though such theories are much more elaborated than their predecessors, they still share the basic logic that limited capacity must be administered and that attention has the job of doing that.

All the approaches that I have discussed so far do not only share the limited-capacity notion but they also consider consciousness, or some philosophically less laden equivalent (like working memory or central processor), as the system that suffers from these limitations and has thus to be saved from overload. A few approaches have questioned this latter implication, however. Authors like Allport (1987) and Neumann (1987) have considered that it may not, or not so much, be conscious representation that constitutes the functionally important bottleneck but our action potentialities. As an example, visual attention may selectively focus on one of many apples on an apple tree not because one's conscious awareness would otherwise be overloaded but, rather, because one can actively pick only one apple at a time anyway. On the one hand, these approaches differ from the main tradition by considering action more important than consciousness, culminating in the claim that selection is for action. On the other hand, however, the limited-capacity notion is not given up, as it is

still scarcity (of action possibilities) that represents the main problem and attention that solves it.

In this chapter, I would like to challenge not only the assumption that attention functions to prevent consciousness from overload (an aim that I share with selection-for-action approaches) but also that the management of scarcity has anything to do with the original biological function of attention. In particular, I will argue that attention is a direct derivative of mechanisms subserving the control of basic motor actions. I'm aware that this is an extreme statement that is likely to require modification in the light of new findings, but at the same time I believe that it can be inspiring and helpful by generating new insights and research questions. To motivate my suggestion, I will first set the theoretical stage by discussing the implications of the primate brain's preference to represent stimulus events and action plans in a distributed, feature-based fashion and to process information concurrently along multiple pathways. Then I will discuss a number of empirical findings that support the general idea that action planning and action control can affect perception and attention and then go on to develop a preliminary theoretical framework that grounds attention in action control.

DISTRIBUTED REPRESENTATIONS AND COMMON CODING

Artificial intelligence, philosophical approaches, and many psychological models assume that the basic units of human cognition can be considered as symbols, so that cognitive processes can be reconstructed as symbol manipulation. Increasing evidence and deeper insights into the structure of the primate cortex suggest a different picture, however. Visual objects, for instance, are known to be coded in terms of their features, which are concurrently analyzed on various feature maps specialized in the processing of orientation, shape, color, motion, and more (DeYoe & Van Essen, 1988). Even at higher representational levels, objects do not seem to be represented by single units but by composites of codes

representing the parts and elements of objects (Tanaka, 2003). This does not rule out the possibility that symbolic representations exist in addition to that, but it does point to the fact that the human brain has a strong tendency to represent perceptual events in a distributed, feature-based fashion. This tendency is not restricted to perceptual coding. Among other things, separate neural networks are coding the direction of an arm movement (Georgopoulos, 1990), its force (Kalaska & Hyde, 1985), and distance (Riehle & Requin, 1989), suggesting that action plans are composites of codes of separately specified action features.

The distributed, feature-based representation of perceptual events and action plans is also reflected in numerous behavioral observations. For instance, searching for a single visual feature (a particular shape, say) in perceptually crowded scenes or arrays is much easier than searching for a feature conjunction (a particular shape in a particular color; Treisman & Gelade, 1980) and if people are to report feature conjunctions under attentionally demanding conditions they tend to fabricate illusory conjunctions (Treisman & Schmidt, 1982). With regard to action planning, different parameters of manual movements can be precued separately and through different stimuli, with the eventual reaction time decreasing as a function of the number of precues (e.g., Rosenbaum, 1980; Lépine, Glencross, & Requin, 1989). Even interactions between stimuli and actions provide evidence for feature-based representations: For instance, stimulus events prime responses, and action plans affect perceptual processes, if and to the degree that stimuli and responses share features, such as location (Hommel et al., 2001a; Kornblum, Hasbroucq, & Osman, 1990).

Especially these latter observations—that stimulus representations and response representations can interact, and that these interactions depend on feature overlap—have important implications with regard to the question of how stimuli and responses are cognitively represented and how these representations are related. According to Hommel et al. (2001a), both perceived events (i.e., stimuli) and to-be-produced events (i.e., action plans) are

represented by cognitive codes of their distal features and, thus, in a common format. These codes are composites of sensorimotor units, which relate perceived action effects to the motoric means employed to produce them (Elsner & Hommel, 2001). According to this logic, seeing a red pen on one's desk, say, is the result of having directed one's eyes, and perhaps even one's head and body, towards the location of the pen, so that the visual information the pen provides is the action effect of these motor movements and will thus be integrated with them. Perceiving and acting is thus the same process, consisting of moving one's body in order to generate particular perceptions. If so, there is no qualitative difference between the representation of a stimulus event (which includes the action that has given rise to it) and the representation of an action plan (which includes the perceptual event the action aims at—the action goal, that is).

If perceptual events and action plans are represented in a common format, and if this format refers to bundles or bindings of perceptual features and motor parameters (Hommel, 2004), one would expect that control processes operating on these cognitive representations have characteristics that reflect this distributed, feature-based format. Indeed, there is increasing evidence that input and output control (i.e., attentional and intentional selection) operates on feature dimensions. For instance, if people search complex visual scenes for visually deviant targets (i.e., stimuli that pop out because of their unique color, shape, etc.), they are better if they can anticipate with regard to which feature dimension an upcoming target will deviate (e.g., Müller, Heller & Ziegler, 1995). This suggests that people can strategically increase the weights or “gain” of a particular feature dimension in order to facilitate the coding of features falling on it (Found & Müller, 1996). The same conclusion is suggested by observations from studies on task switching. In such studies, subjects often carry out responses to stimuli that are defined by one of multiple feature dimensions, such as to the color vs. the meaning of colored color words (Allport, Styles, & Hsieh, 1994), or to the

horizontal vs. vertical location of stimuli (e.g., Meiran, 1996). Performance is much better if the task-relevant feature dimension is repeated than if it is alternated, suggesting that switching between different task sets takes time and effort. Importantly for our purposes, implementing a new task set is assumed to include directing attention to the target-defining stimulus dimension (Logan & Gordon, 2001) and the response-defining action dimension (Meiran, 2000). That is, executive control operates on feature dimensions, presumably by altering the weights that determine the degree to which features coded on these dimensions are considered by, or affect, cognitive processes.

MULTIPLE PROCESSING PATHWAYS

There is increasing evidence that the human brain does not only code perceived and produced events in a distributed fashion but that it also concurrently processes different aspects of events along different neural pathways. One of the best-known distinctions between parallel processing codes is that between the dorsal and the ventral pathway (Ungerleider & Mishkin, 1982). Early approaches have characterized these two pathways in terms of where-versus what-processing. Whereas the dorsal pathway was considered to process spatial attributes of perceived events, the ventral pathway was thought to process identity-related attributes, such as shape and color. Later approaches, Milner and Goodale (1995) in particular, have suggested an alternative interpretation in terms of action-related (or pragmatic) processing versus perception-related processing. That is, the dorsal pathway was considered to directly feed into action control, without being accessible for conscious perception, whereas the ventral pathway was thought to mainly subserve conscious and unconscious perceptual processes. In view of increasing evidence that is not quite consistent with this particular subdivision, recent reformulations have suggested an interpretation in terms of online control of action—attributed to the dorsal pathway—versus action planning—

a presumably ventral activity (Glover, 2004; Hommel, Müsseler, Aschersleben, & Prinz, 2001b).

Interestingly, these neuroscientifically motivated considerations fit well with theoretical developments in the domain of action planning and control. Modern cognitive approaches were driven by the insight that human action is commonly goal driven and must, thus, be controlled by some kind of internal representation (Lashley, 1951). Authors like Keele (1968) have pushed this possibility to an extreme and assumed that all muscle parameters and commands of a movement are stored and used to construct motor programs that prestructure all aspects of a movement in advance. Others, however, have pointed out that this possibility would put too high demands on storage and render action planning very inflexible, as each slight change of a movement would require a separate program (Schmidt, 1975). Theoretically more reasonable are hybrid approaches that assume that only some, structural or invariant features of an action are stored and used for later programming, whereas more variable features are specified by online information (e.g., Schmidt, 1975). Consistent with this consideration, studies have shown that transferring from one task to another is easier if the two tasks share invariant features, whereas changes in variant features do not affect performance much (see Heuer, 1991).

Behavioral and neuroscientific approaches thus converge on the idea that action control is comprised of two processes: *action planning*, which consists of specifying the basic structure of an action, including its most relevant, invariant features and can be performed online as well as off-line (i.e., some time before the action is executed), and online *action adjustment*, which consists of fine-tuning the action by specifying the remaining features and open parameters. A particularly elegant illustration of the interplay between action planning and action adjustment is provided by studies using the so-called double-step paradigm. For instance, in a study by Prablanc and Pélisson (1990), subjects were asked to move their right

index finger from a home position to a light spot, and the spatial and temporal parameters of the movement were measured. In some trials, the target spot was moved a little further away from the subject while he or she was already moving. Importantly, the target was moved during an eye blink, so that subjects were unable to see the change. The most relevant outcome was that, first, the finger correctly reached the target even in change trials and that, second, this was achieved without any measurable hesitation of the moving hand. In a manner of speaking, the hand was smarter, better informed, and more adaptive than the mind. So, even though we can assume that goal-directed reaching movements are prepared and programmed in advance, a slight change in the location of the target does not require time-consuming modifications of the program or complete reprogramming. This means that the original program did not include specific information about the target location but left the specification of the details to online routines that adjusted the action on the fly.

Distributing the labor over different processing channels has obvious advantages: storage and preplanning is minimized and yet the resulting action is as precise as necessary. However, just like distributed representations create binding problems (Treisman, 1996), distributed processing creates coordination problems. In one way or another, action-planning processes need to inform action-adjustment processes about which parameters to fill or specify, and how to do so. For instance, Milner and Goodale (1995) claim that their dorsal action pathway does not have any memory capacity and does not interact with, nor is informed by, ventrally-mediated, conscious or unconscious decision making. This would imply that the channel that is dedicated to action control has no way to plan any action, retrieve or access any action plan, and cannot have any idea about currently relevant action goals. It is difficult to see how such a channel can do the job it is supposed to do: to select relevant sensory features and feed them into the action programs. Obviously, coherent, goal-

directed action requires some kind of coordination between planning and adjustment processes, so that the latter can provide what the former leave open.

This chapter is devoted to this kind of coordination problem, and I will present a principled approach to how it might be solved. An important insight pointing to a possible solution is that concurrent processing streams need to be conditionalized by the current action goal. Action goals, so I will assume, govern the selection and planning of appropriate actions, and this planning process biases concurrent processing streams, such as the one in charge of action adjustments, towards information that is suitable to specify the action parameters that planning processes left open. A particularly interesting implication of this line of thought is that it requires action-related processes to affect perception and attention to perceptual input. Indeed, as the next section shows, there are numerous findings suggesting that action planning does affect perception and attention.

ACTION CONTROL AND ATTENTION

An early suggestion that visual attention may be affected by action planning emerged from studies on the so-called meridian effect (Rizzolatti, Riggio, Dascola, & Umiltà, 1987). This effect can be observed in studies that use attentional cues. Consider, for instance, a subject focusing on a central spot in a visual display, which further consists of four possible target locations marked by small frames, two at the left and two at the right of fixation. Now assume that, in each trial, one of the four locations is precued with high validity—that is, the subject knows in which of the four frames the target is likely to appear. If the target then actually appears in the precued frame, reaction times can be expected to be fast, suggesting that subjects “moved their attention” to the frame (Posner, 1980). But what if the target appears in an uncued frame? As Rizzolatti et al. (1987) observed, reaction times are not only slower in this case but depend on the spatial relation between the cued frame and the eventual target location. If, for instance, one of the two inner frames was cued, performance was better

if the target frame was located on the same side of the cue than on the opposite side. In other words, moving attention further into the same direction was less costly than changing the direction. According to the authors, this may suggest that attention is moved by programming (but not necessarily executing) an eye movement, which may require the sequential specification of a direction parameter and a distance parameter—in this order. If the direction stays the same (as when, say, the inner left frame is cued but the target appears in the outer left frame), only the distance parameter needs to be modified, which can be done faster than modifying the direction parameter or both parameters.

Further evidence for the general idea that the programming of eye movements is involved in directing visual attention to locations in space (see also Klein, 1980) stems from Deubel and Schneider (1996; Schneider & Deubel, 2002). Their subjects were to carry out saccades to visual targets on the left or right of a fixation point. Before moving their eyes they were briefly flashed with a visual string of stimuli containing a to-be-discriminated target symbol. As it turned out, performance was good only if the location of the visual target coincided with the goal of the saccade, suggesting that programming the saccade involves moving attention to the goal location in advance of the saccade—which then facilitates the processing of stimuli appearing there. These observations are consistent with the premotor theory of attention but go beyond previous findings in directly demonstrating that saccade programming actually matters for spatial selection.

Interactions between the programming of eye movements and attentional selection support the idea that action planning affects attentional control, but they are too restricted to provide a basis for a comprehensive action-based theory of attention. First, even though linking overt and covert visual attention (i.e., attending by moving the physical versus the “mind’s” eye) has a long tradition in psychology (e.g., James, 1890; Posner, 1980), this may be due to the particularly strong and straightforward sub-cortical connections between retinal

input processing and movements of the eyeballs. This raises the question of whether other than oculomotor action planning can affect attention. Second, the observed interactions between action and attention were restricted to spatial selection. Even though the spatial selection of relevant information plays an important role in perception and action, human attention subserves more functions than that—just think of object-based selection, action selection, and integration (Schneider, 1995). Fortunately, however, there is increasing evidence of interactions between manual and verbal action planning and attentional functions other than spatial selection.

First evidence for the impact of manual action planning on visual processing was provided by Müsseler and colleagues. Müsseler and Hommel (1997), for instance, had participants prepare a left- or right-hand key press and carry it out whenever they felt ready. To signal their readiness they pressed a spatially neutral readiness key before performing the prepared action. Pressing the readiness key triggered the presentation of a masked visual arrowhead that pointed to the left or right. At the end of the trial, participants reported at leisure in which direction the arrowhead pointed, which, given the masking procedure, was difficult and attention-demanding. The important observation was that the accuracy of the perceptual report was dependent on the relation between the prepared response and the direction of the arrowhead. If participants prepared and carried out a left-hand response, they had substantially more difficulty detecting a left-pointing than a right-pointing arrowhead, and the opposite was true for right-hand responses. In other words, planning a spatially defined manual action “blinded” the participants to perceptual events that shared features with the action.

Even though this finding seems counterintuitive, it fits with the idea that action planning consists in the binding of distributed feature codes that specify the action’s relevant characteristics (Stoet & Hommel, 1999). Planning a left-hand action would thus require the

binding of a <left> code with other relevant codes specifying, say, the speed, force, and extent of the key press. If we further assume that perceptual and action-related features are coded in the same format (Hommel et al., 2001a; Prinz, 1990), “occupying” (Stoet & Hommel, 1999) a given feature by binding it into an action plan should indeed impair the creation of another binding to represent a feature-overlapping perceptual event—such as a spatially compatible arrowhead. Other observations confirmed that this line of reasoning is not restricted to manual action plans or spatial relationships. For instance, planning a manual left or right action “blinds” participants to compatible left- or right-pointing arrowheads but not to the words “left” or “right”, whereas planning a vocal action (i.e., saying aloud “left” or “right”) impairs the perception of compatible words but not arrowheads (Hommel & Müsseler, 2006).

Another demonstration of interactions between manual action planning and visual attention was provided by Craighero, Fadiga, Rizzolatti, and Umiltà (1999). They had participants manually grasp invisible objects that were tilted to the left or right. The type of grasp was planned ahead but the execution had to await the presentation of a go signal. The orientation of this go signal did or did not match the orientation of the to-be-grasped object. It turned out that participants were responding faster if the invisible target object and the go signal matched in orientation (and even if the go signal was responded to by foot), suggesting that planning a grasping action prepared the visual system for the processing of target-related features. Similarly, Bekkering and Neggers (2002) had participants detecting and grasping (versus pointing to) visual targets defined by a conjunction of orientation and color features. The findings revealed that fewer orientation errors were committed when participants prepared for grasping as compared to pointing, whereas color errors were rare in all conditions. The authors argue that planning a particular movement enhances the processing of features that specify the target of this movement. At first sight, these observations do not seem to fit with the inverse effect on feature overlap reported by Müsseler and Hommel

(1997). However, while Müsseler and Hommel required participants to consciously perceive and report the perceptual events, participants in the Craighero et al. and the Bekkering and Neggers studies were only using these events for triggering a more or less prepared response—a situation that is unlikely to require feature binding.

Let us summarize so far. The apparent distribution of labor between off-line action-planning processes and online action adjustment introduces a control problem and raises the question of how action planning can make sure that adjustment processes select the appropriate sensory information and feed it into the relevant motor-control structures. We have seen a number of empirical phenomena suggesting that planning an action has a direct impact on attentional and perceptual processes, and we have also seen that this holds for oculomotor, manual, and vocal actions, and corresponding perceptual dimensions. In principle, it thus seems possible that action planning processes do not only specify the task-relevant characteristics of a given action but that they also bias action-adjustment routines towards the relevant perceptual dimensions. And yet, there is one fly in the ointment: whereas research on visual attention suggests that task goals lead to the priming and stronger weighting of appropriate perceptual *dimensions*, at least most of the available evidence for action-attention interactions points to stimulus-specific biases (e.g., the priming of one particular orientation in Craighero et al., 1999). The theoretical challenge thus consists in explaining why and how action planning can bias perceptual processing towards perceptual dimensions that provide information for specifying the open parameters of the action in question.

INTENTIONAL CONTROL OF ATTENTION: A NEW FRAMEWORK

The theoretical framework I would like to propose here was motivated by an observation of Schubotz and von Cramon (2001, 2002). They had participants carry out an oddball task while lying in an fMRI scanner. Sequences of stimuli that followed a particular

rule were presented (e.g., a repeated sequence of particular colors, locations, or shapes), and the participant was to report at leisure at the end of the trial whether one of the stimuli violated the rule. The important observation was that this perceptual monitoring task consistently activated the lateral premotor cortex, even in the absence of any motoric response. A meta analysis of these and similar observations revealed systematic relationships between the task-relevant perceptual dimension and the particular area in the premotor cortex where the activation was located (Schubotz & von Cramon, 2003). Three of these relations were particularly systematic: Location-relevant perceptual monitoring engaged premotor areas that are involved in the control of saccades and reaching movements; the monitoring of object-related features (such as color or shape) activated premotor areas involved in the control of grasping movements; and the monitoring of rhythmic events engaged premotor areas responsible for controlling vocal actions and manual tapping. As the authors point out, these relationships suggest that action-related brain areas are directly involved in the control of attention and, in particular, in directing attention towards action-related perceptual dimensions.

These considerations were further developed by Fagioli, Hommel, and Schubotz (2007a). Preparing for a reaching movement, these authors reasoned, should sensitize the perceptual system for features of dimensions that are relevant for specifying the open parameters of reaching movements. Most likely, this criterion is met by location information. Preparing for a grasping movement, in turn, should sensitize the system for processing information about the final phase of the grasp, such as the size of the object signaling the hand's aperture. To test these hypotheses, the authors had participants reach towards or grasp an object in front of them. Before the action was executed, however, participants were presented with a sequence of stimuli following a particular rule, as in the setup of Schubotz and von Cramon (2001), and they were to detect possible oddballs. If an oddball occurred, the

prepared reaching or grasping movement was carried out. As expected, the reaction times for these movements varied with the perceptual dimension on which the oddball was defined. Whereas reaching movements were initiated faster with location oddballs than with size oddballs, the opposite applied to grasping movements. To rule out that this effect was due to the oddball-induced priming of the movement, another experiment was carried out in which the detection of the oddball was signaled by a foot response. Again, preparing for a reaching movement facilitated the detection of location oddballs, and preparing for a grasping movement facilitated the detection of size oddballs.

These observations are consistent with the idea that action control encompasses the priming of perceptual dimensions, but one may argue that this connection is less direct than suggested here. For instance, it may be that a general executive control system does not only select appropriate responses but also implements a particular attentional set. Indeed, Logan and Gordon (2001) have suggested that executive control functions both bias attention towards task-relevant perceptual dimensions and specify the necessary stimulus-response rules without directly relating these two processes to each other or even deriving the attentional bias from action-control demands. In an attempt to provide more specific evidence for action-induced attentional biases, Fagioli, Ferlazzo, and Hommel (2007b) investigated whether the biases observed by Fagioli, Hommel, and Schubotz require active action planning. If activating an action plan is sufficient to induce the stronger weighting of related perceptual dimensions, they reasoned, such weighting should also be observed if the action plan is activated involuntarily. Participants were again monitoring sequences of stimuli and were to press a foot pedal as soon as they detected an oddball. They did not carry out any other action and no reaching or grasping movement in particular. However, prior to the stimulus sequences short video clips were presented, which showed a person carrying out a reaching or grasping movement. These videos were not relevant to the task and did not predict

or inform about the stimulus sequences or the correct responses. Nevertheless, participants were faster to detect location oddballs after seeing a reaching movement and size oddballs after seeing a grasping movement. Apparently, the videos activated reaching- and grasping-related action plans and this activation was sufficient to increase the weights of reaching- and grasping-related perceptual dimensions.

=== FIGURE 1 ===

Taken together, these findings support the idea that the mere activation of an action plan—whether through top-down processes in the service of the current action goal or bottom-up, stimulus-induced processes—leads to an increase on the weights of those perceptual dimensions that allow for the specification of action parameters commonly left open by action planning. Figure 1 summarizes the theoretical implications of this consideration. As pointed out above, stimuli are assumed to be coded on feature maps, with each feature activating a code on the respective feature dimension (or multiple codes competing for coding the stimulus: Reynolds, Chelazzi, & Desimone, 1999). In the example given, a circular object at some top location is coded on a shape and a location map—a drastic simplification that is not meant to deny the existence of numerous other feature maps (such as color, motion, etc.), of multiple spatial maps (coding for, e.g., allocentric, egocentric, and retinal location), and of other sensory modalities.

This information is propagated to two different processing pathways, one subserving perception and action planning (similar but not identical to the ventral pathway of Milner & Goodale, 1995, and comparable to the action-planning pathway of Glover, 2004) and one subserving online action adjustments (comparable to Milner and Goodale's dorsal pathway and Glover's action-control pathway). Activating an action plan, such as for grasping or reaching (symbolized by the grasping and pointing hands in the figure), increases the weight (ω) of the output of particular feature maps, which increases the impact of information coded

there on further information processes (i.e., perception and action planning on the one hand and action adjustment on the other). Following the Theory of Event Coding (Hommel et al., 2001), perception and action planning are not further differentiated, which acknowledges that these two functions highly interact and can be considered two sides of the same coin.

Perception and action planning creates action plans reflecting the current goal. Action plans consist of specified parameters (structural features of the planned action that are relevant for reaching the goal) and not-yet-specified parameters that are to be filled by online adjustment processes; in the figure, these parameters are symbolized by black and white circles, respectively. The open parameters are specified by continuously transmitting sensory information from feature maps to ongoing actions. This transmission is weighted by the output weight ω of the respective dimensions. Accordingly, given that planning a grasping action increases the weights for shape information, action-adjustment processes will mainly consider information provided by the shape map and use it to specify the remaining grasping parameters (such as hand aperture).

Note that the output weights have two functions in this model. On the one hand, they help to overcome the control problem posed by the existence of multiple concurrent processing streams by biasing online adjustment processes towards goal-relevant perceptual dimensions. On the other hand, they also bias perception and action planning towards these dimensions, a characteristic that is important to account for the findings of Fagioli et al. (2007a, 2007b). These findings suggest that planning an action leads to the faster conscious detection of stimuli varying on action-relevant perceptual dimensions, which implies that planning must have affected perceptual processes. Recent findings support the idea that action planning modulates conscious perception in systematic ways. Wykowska, Schubö, and Hommel (2008) presented participants with visual-search displays that contained to-be-detected pop-out targets—that is, stimuli that differed from all other stimuli of the display on

one dimension. As in the studies of Fagioli and colleagues, participants prepared either a reaching or a grasping action, and the target-defining dimensions were luminance (which was considered more important for reaching than for grasping) and size (which was considered more important for grasping than for reaching). If participants knew in advance on which perceptual dimensions a target would pop out, the prepared action biased attention systematically: preparing a reach facilitated the detection of luminance-defined targets and preparing a grasp the detection of size-defined targets. However, this effect was not observed when participants did not know the target-defining dimension in advance. Under this kind of uncertainty participants are known to not prepare for a particular perceptual dimension but to rely on saliency signals—that is, they respond to any dishomogeneity in the visual field without identifying the dimension on which it occurs (Bacon & Egeth, 1994). Given the absence of action-induced biases on this type of processing, it makes sense to assume that these biases target the output of feature-map coding, but not the input or processes preceding or circumventing feature coding.

THEORETICAL IMPLICATIONS

The proposed theoretical framework has a number of interesting theoretical implications that break with the main line of reasoning underlying traditional attentional research. Most importantly, it denies that attentional functions emerged to distribute sparse cognitive resources to prevent the cognitive system from overload. In contrast, it proposes that attentional functions originally evolved to deal with control problems arising from distributed representation and processing and from the share of labor between off-line, anticipatory action planning and online action adjustment in particular. Once these functions were available, they could also be used for other purposes—that is, the weights of perceptual dimensions could be manipulated for other reasons than action adjustment and without actually preparing overt actions. It is this generalization that makes people good performers in visual-search

experiments and related tasks. However, outside of the psychological laboratory there are not too many occasions in which selective attention is needed for other purposes than action control—we commonly do not detect feature conjunctions in complex visual environments for the sake of detecting them but do so in the service of particular action goals. Considering this, selection-for-action approaches (Allport, 1987; Neumann, 1987) go in the right direction in emphasizing the theoretical importance of actions. However, the available evidence allows for an even more radical interpretation, according to which attentional functions do not only consider action opportunities but may be a mere byproduct of action control in a distributed processing system.

Given the systematic interactions between particular types of actions and particular perceptual dimensions, it is interesting to ask where this systematicity comes from. One possibility is phylogenetic development—that is, the discovery that some perceptual dimensions are more important for some actions than others may be an evolutionary achievement that became genetically coded over time. Alternatively, the selective use of perceptual dimensions may be an ontogenetic discovery. Consider, for instance, a learning process that is sensitive to the success of actions. In the beginning, actions may be carried out on the basis of any available information, with a noisy and random weighting of information provided by the available feature maps and very mixed results. The open parameters of an action—a grasp, say—would thus be randomly filled with all sorts of feature information. However, extensive experience will reveal that using size information renders grasping actions more successful than, say, using color information, and correlation learning would be sufficient to detect the relationship between different perceived sizes of the grasped object and the hand aperture in the final phase of a grasp. In other words, exploration in infancy and early childhood may allow for the discovery of optimal relationships between the

consideration of particular perceptual dimensions on the one hand and particular action categories on the other.

Let us conclude by considering the implications of this suggested framework for the topic of this book, the issue of whether and when attentional processes are effortful or effortless. According to the suggested framework, attentional operations themselves are not effortful but are more or less automatically triggered by action-control processes, which again are coordinated by the current action goal (see Fagioli et al., 2007b). Hence, the selection, representation, and maintenance of an action goal would be a necessary precondition for attentional processes to operate, and these processes are commonly considered effortful. The most common task to investigate goal implementation requires participants to switch between different, mutually incompatible action goals (e.g., Monsell, 2003). Using this task has revealed two major findings that are important for our purposes. First, performance is strongly impaired in trials that require a goal switch, which has been taken to reflect time demands associated with establishing the new goal before going on with the task details (e.g., Rogers & Monsell, 1995; Meiran, 1996). Second, even task repetitions have been found to show performance decreases over time, suggesting that goal maintenance requires some effort (e.g., Altmann, 2002). Even though such observations seem to make a strong case for effortful goal operations, there are reasons to not jump to conclusions. Waszak, Hommel, and Allport (2003) provided evidence that task goals can become associated with particular stimuli, so that these stimuli can act as exogenous retrieval cues for these goals. Along the same lines, Logan and Bundesen (2003) observed that most of the difficulty in switching between different goals is due to a shift in the task cues that signal the different goals—again suggesting that goal selection can become stimulus driven under appropriate circumstances. Indeed, Bargh and Gollwitzer (1994; Bargh, Gollwitzer, Chai, Barndollar, & Troetschel, 2001) have claimed that everyday behavior is often driven by external cues, which would

allow for effortless goal selection. Similarly, even if goal maintenance turned out to require effort in artificial laboratory tasks, the goals we maintain in everyday life are commonly consistent with, and thus supported by, long-term motives and overarching goals as well as by environmental cues. Indeed, situations in which the available stimuli are specifically associated with different tasks, switching between tasks and goals was not found to be effortful or performance-costly (Jersild, 1927). Taken altogether, it may thus be possible that the frequent use of artificial tasks that are not deeply anchored in the participant's motivational structure and not supported by environmental cues have lead to a rather drastic overestimation of the cognitive effort needed to deal with everyday life.

REFERENCES

Allport, A. (1987). Selection for action: some behavioral and neurophysiological considerations of attention and action. In H. Heuer & A. F. Sanders (Eds.), *Perspectives on Perception and Action* (pp. 395-419). Hillsdale, NJ: Lawrence Erlbaum Associates.

Allport, D. A. (1993). Attention and control. Have we been asking the wrong questions? A critical review of twenty-five years. In D. E. Meyer & S. Kornblum (Eds.), *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience* (pp. 183–218). Cambridge, MA: MIT Press.

Allport, A., Styles, E. A., & Hsieh, S. (1994). Shifting intentional set: Exploring the dynamic control of tasks. In C. Umiltà, & M. Moscovitch (Eds.), *Attention and performance XV* (pp. 421-452). Cambridge, MA: MIT Press.

Altmann, E. M., (2002). Functional decay of memory for tasks. *Psychological Research*, 66, 287-297.

Bacon, W. F., & Egeth, H. E. (1994). Overriding stimulus-driven attentional capture. *Perception and Psychophysics*, 55, 485-496.

Bargh, J. A., & Gollwitzer, P. M. (1994). Environmental control over goal-directed action. *Nebraska Symposium on Motivation*, 41, 71-124.

Bargh, J. A., Gollwitzer, P. M., Chai, A. L., Barndollar, K., & Trötschel, R. (2001). Automated will: Nonconscious activation and pursuit of behavioral goals. *Journal of Personality and Social Psychology*, 81, 1014-1027.

Bekkering, H., & Neggers, S.F.W, (2002). Visual search is modulated by action intentions. *Psychological Science*, 13, 370-374.

Broadbent, D. E. (1958). *Perception and communication*. London: Pergamon.

Bundesen, C., Habekost, T., & Kyllingsbaek, S (2005). A neural theory of visual attention. Bridging cognition and neurophysiology. *Psychological Review*, *112*, 291-328.

Craighero, L., Fadiga, L., Rizzolatti, G., Umiltà, C., A. (1999). Action for perception: a motor-visual attentional effect. *Journal of Experimental Psychology: Human Perception and Performance*, *25*, 1673-1692.

Deubel, H., & Schneider, W.X. (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research*, *36*, 1827-1837.

DeYoe, E. A., & Van Essen, D. C. (1988). Concurrent processing streams in monkey visual cortex. *Trends in Neuroscience*, *11*, 219-226.

Elsner, B., & Hommel, B. (2001). Effect anticipation and action control. *Journal of Experimental Psychology: Human Perception and Performance*, *27*, 229-240.

Fagioli, S., Ferlazzo, F., & Hommel, B. (2007b). Controlling attention through action: Observing actions primes action-related stimulus dimensions. *Neuropsychologia*, *45*, 3351-3355.

Fagioli, S., Hommel, B., & Schubotz, R. I. (2007a). Intentional control of attention: Action Planning primes action related stimulus dimensions. *Psychological Research*, *71*, 22-29.

Found, A., & Müller, H. J. (1996). Searching for unknown feature targets on more than one dimension: Investigating a 'dimension weighting' account. *Perception and Psychophysics*, *58*, 88-101.

Georgopoulos, A. P. (1990). Neurophysiology of reaching. In M. Jeannerod (Ed.), *Attention and Performance XIII: Motor representation and control* (pp. 227-263). Hillsdale, NJ: Erlbaum.

Glover, S. (2004). Separate visual representations in the planning and control of action. *Behavioral and Brain Sciences*, *27*, 3-78.

Heuer, H. (1991). Invariant relative timing in motor-program theory. In J. Fagard & P. H. Wolff (Eds), *The development of timing control and temporal organization in coordinated action* (pp. 37-68). Amsterdam: North-Holland.

Hommel, B. (2004). Event files: feature binding in and across perception and action. *Trends in Cognitive Sciences*, 8, 494-500.

Hommel, B., & Müsseler, J. (2006). Action-feature integration blinds to feature-overlapping perceptual events: Evidence from manual and vocal actions. *Quarterly Journal of Experimental Psychology*, 59, 509-523.

Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001a). The Theory of Event Coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24, 849-937.

Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001b). Codes and their vicissitudes. *Behavioral and Brain Sciences*, 24, 910-937.

James, W. (1890). *The principles of psychology*. New York: Dover Publications.

Jeannerod, M. 1984. The timing of natural prehension movements. *Journal of Motor Behavior*, 16, 235-254.

Jersild, A. T. (1927). Mental set and shift. *Archive of Psychology*, whole no. 89.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice-Hall.

Kalaska, J. F., & Hyde, M. L. (1985). Area 4 and area 5: Differences between the load direction-dependent discharge variability of cells during active postural fixation. *Experimental Brain Research*, 59, 197-202.

Keele, S. W. (1968). Movement control in skilled motor performance. *Psychological Bulletin*, 70, 387-403.

Klein, R. (1980). Does oculomotor readiness mediate cognitive control of visual attention? In R. S. Nickerson (Ed.), *Attention and performance VIII* (pp. 259-276). Hillsdale, NJ: Erlbaum.

Kornblum, S., Hasbroucq, T., & Osman, A. (1990). Dimensional overlap: Cognitive basis of stimulus-response compatibility—A model and taxonomy. *Psychological Review*, *97*, 253-170.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior* (pp. 112-146). New York: John Wiley & Sons.

Lépine, D., Glencross, D., & Requin, J. (1989). Some experimental evidence for and against a parametric conception of movement programming. *Journal of Experimental Psychology: Human Perception and Performance*, *15*, 347-362.

Logan, G. D., & Bundesen, C. (2003). Clever homunculus: Is there an endogenous act of control in the explicit task-cuing procedure? *Journal of Experimental Psychology: Human Perception and Performance*, *29*, 575-599.

Logan, G. D., & Gordon, R. D. (2001). Executive control of visual attention in dual-task situations. *Psychological Review*, *108*, 393-434.

Meiran, N. (1996). Reconfiguration of processing mode prior to task performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 1-20.

Meiran, N. (2000). Modeling cognitive control in task-switching. *Psychological Research*, *63*, 234-249.

Milner, A. D., & Goodale, M. A. (1995). *The visual brain in action*. Oxford: Oxford University Press.

Monsell, S. (2003). Task switching. *Trends in Cognitive Sciences*, *7*, 134-140.

Müller, H. J., Heller, D., & Ziegler, J. (1995). Visual search for singleton feature targets within and across feature dimensions. *Perception and Psychophysics*, *57*, 1-17.

Müsseler, J., & Hommel, B. (1997). Blindness to response-compatible stimuli. *Journal of Experimental Psychology: Human Perception and Performance*, *23*, 861-872.

Neumann, O. (1987). Beyond capacity: A functional view of attention. In H. Heuer & A. F. Sanders (Eds), *Perspectives on perception and action* (pp. 361-394). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Posner, M., I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, *32*, 3-25.

Prablanc, C., & Pélisson, D. (1990). Gaze saccade orienting and hand pointing are locked to their goal by quick internal loops. In M. Jeannerod (Ed.), *Attention and performance XIII* (pp. 653-676). Hillsdale, NJ: Erlbaum.

Prinz, W. (1990). A common coding approach to perception and action. In O. Neumann & W. Prinz (Eds.), *Relationships between perception and action: Current approaches* (pp. 167-201). Berlin: Springer.

Reynolds, J. H., Chelazzi, L., & Desimone, R. (1999). Competitive mechanism subserve attention in macaque areas V2 and V4. *Journal of Neuroscience*, *19*, 1736-1753.

Riehle, A. & Requin, J. (1989). Monkey primary motor and premotor cortex: Single-cell activity related to prior information about direction and extent of an intended movement. *Journal of Neurophysiology*, *61*, 534-549.

Rizzolatti, G., Riggio, L., Dascola, I., & Umiltà, C (1987). Reorienting attention across the horizontal and vertical meridians: Evidence in favor of a premotor theory of attention. *Neuropsychologia*, *25*, 31-40.

Rogers, R. D., & Monsell, S. (1995). Cost of a predictable switch between simple cognitive tasks. *Journal of Experimental Psychology: Human Perception and Performance*, *124*, 207-231.

Rosenbaum, D. A. (1980). Human movement initiation: Specification of arm, direction and extent. *Journal of Experimental Psychology: General*, *109*, 444-474.

Schmidt, R. A. (1975). A schema theory of discrete motor skill learning. *Psychological Review*, *82*, 225-260.

Schneider, W. X. (1995). VAM: A neuro-cognitive model for visual attention control of segmentation, object recognition, and space-based motor action, *Visual Cognition*, *2*, 331-375.

Schneider, W. X. & Deubel, H. (2002). Selection-for-perception and selection-for-spatial-motor-action are coupled by visual attention: A review of recent findings and new evidence from stimulus-driven saccade control. In W. Prinz & B. Hommel (Eds), *Attention and Performance XIX: Common Mechanisms in Perception and Action* (p. 609-627). Oxford: Oxford University Press.

Schubotz, R., I., & von Cramon, D. Y. (2001). Functional organization of the lateral premotor cortex: fMRI reveals different regions activated by anticipation of object properties, location and speed. *Cognitive Brain Research*, *11*, 97-112.

Schubotz, R., I., & von Cramon, D. Y. (2002). Predicting perceptual events activates corresponding motor schemes in lateral premotor cortex: An fMRI study. *Neuroimage*, *15*, 787-796.

Schubotz, R., I., & von Cramon, D. Y. (2003). Functional-anatomical concepts of human premotor cortex: Evidence from fMRI and PET studies. *Neuroimage*, *20*, S120-S131.

Stoet, G., & Hommel, B. (1999). Action planning and the temporal binding of response codes. *Journal of Experimental Psychology: Human Perception and Performance*, *25*, 1625-1640.

Tanaka, K. (2003). Columns for complex visual object features in the inferotemporal cortex: clustering of cells with similar but slightly different stimulus selectivities. *Cerebral Cortex*, *13*, 90-99.

Treisman, A. (1988). Features and objects: The Fourteenth Bartlett Memorial Lecture. *The Quarterly Journal of Experimental Psychology*, *40A*, 201-237.

Treisman, A. (1996). The binding problem. *Current Opinion in Neurobiology*, *6*, 171-178.

Treisman, A., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*, 97-136.

Treisman, A., & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. *Cognitive Psychology*, *14*, 107-141.

Ungerleider, L. G., & Mishkin, M. (1982). Two cortical visual systems. In: Ingle, D. J., Goodale, M. A. and Mansfield, R. J. W. (Eds.). *Analysis of Visual Behaviour* (pp. 549-586). Cambridge, MA: MIT Press.

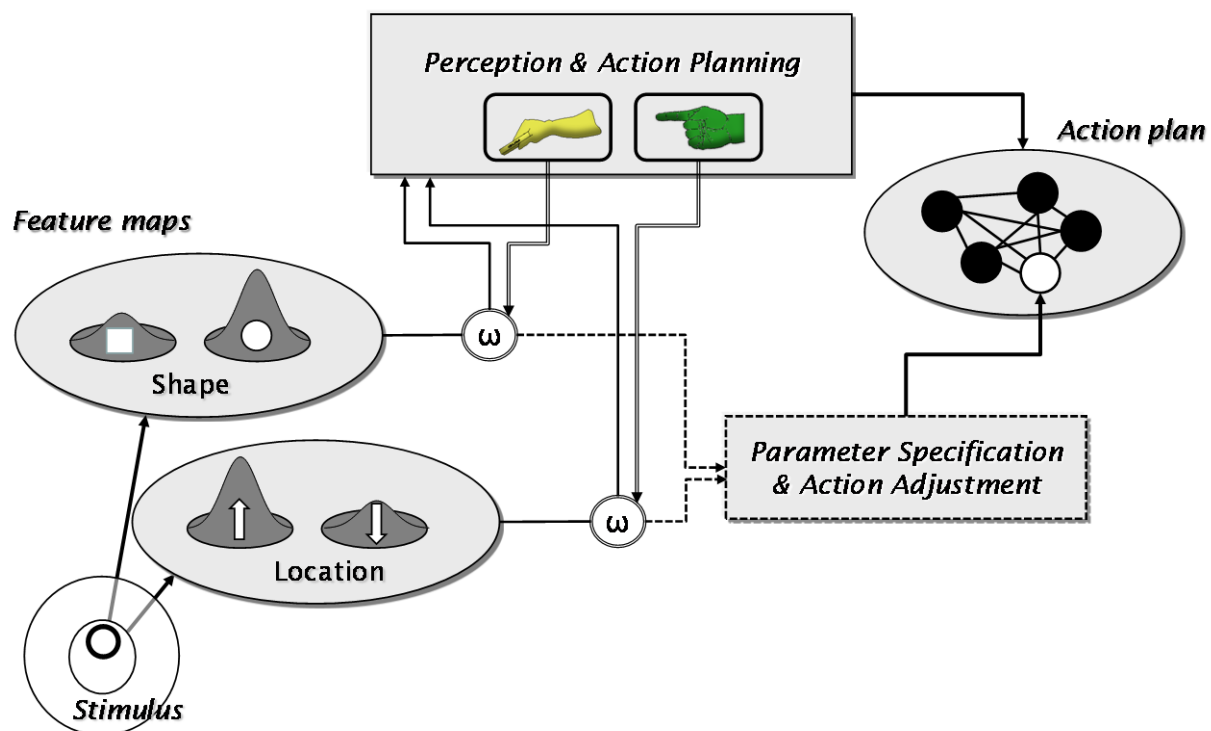
Waszak, F., Hommel, B., & Allport, A. (2003). Task-switching and long-term priming: Role of episodic stimulus-task bindings in task-shift costs. *Cognitive Psychology*, *46*, 361-413.

Wolfe, J. M. (1994). Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, *1*, 202-238.

Wykowska, A., Schubö, A., & Hommel, B. (2008). How you move is what you see: Action planning biases selection in visual search. Submitted.

FIGURE CAPTION

Figure 1: A process model of action-induced attention, see text for explanation.



When an object is more than a binding of its features: Evidence for two mechanisms of visual feature integration

Bernhard Hommel and Lorenza S. Colzato

*Leiden University Institute for Psychological Research, and Leiden Institute
for Brain and Cognition, Leiden, The Netherlands*

People spontaneously integrate the features of visual events into episodic structures that are reactivated if their ingredients match aspects of the current input. Feature integration has been attributed to either the detection of feature conjunctions or the ad hoc binding of feature codes (e.g., by neural synchronization). We report evidence suggesting that both kinds of integration mechanisms coexist. Replicating earlier findings, repeating one visual feature facilitated performance but only if other visual features were also repeated. However, this effect was more pronounced with real objects as compared to arbitrary combinations of shapes and colours. Moreover, the real-object effect was restricted to visual feature integration but did not extend to visuomotor integration, suggesting that the underlying mechanism subserves perception only. We suggest a dual-process feature-integration model that distinguishes between ad hoc binding, which operates on any possible combination of features alike, and conjunction detection, which selectively operates on familiar feature combinations.

Keywords: Object perception; Event file; Feature integration.

Primate brains represent many aspects of the objects and events they perceive and produce in a distributed, feature-based fashion. The human visual cortex, for instance, consists of various neural maps that code for different visual features of perceived objects, such as orientation, shape, or motion (see DeYoe & van Essen, 1988), and the frontal cortex houses maps coding for the direction, distance, and force of intentional actions (see Hommel & Elsner, in press). These observations have been taken to imply various integration or binding problems (e.g., Treisman, 1996), as they raise

Please address all correspondence to Bernhard Hommel, Leiden University, Department of Psychology, Cognitive Psychology Unit, Postbus 9555, 2300 RB Leiden, The Netherlands. E-mail: hommel@fsw.leidenuniv.nl

© 2008 Psychology Press, an imprint of the Taylor & Francis Group, an Informa business
<http://www.psypress.com/viscog> DOI: 10.1080/13506280802349787

the question how the brain knows which of the feature codes activated at a given time are related to the same event.

Authors differ with respect to how serious they consider these problems to be and how easily they think these problems can be solved. For instance, some authors have claimed that serially operating attentional mechanisms are required to bind related information together (e.g., Treisman & Gelade, 1980), whereas others assume that the retinotopic organization of visual maps provides sufficient information for properly integrating at least visual features (van der Heijden, 1995). Most authors have focused on one of two neural principles that may mediate feature integration and help solving binding problems. One principle is that of *convergence*: Lower level neurons may code for simple features, such as <round> or <green>, and project onto higher level neurons that code for feature conjunctions (e.g., <round> AND <green>). This may lead up to even higher level representations of whole objects (Barlow, 1972). Given the considerable variability of objects in terms of their instances and retinal projections, as well as the numerous ways in which features can be potentially combined, the exclusive reliance on convergence mechanisms would lead to a combinatorial explosion and is therefore not particularly plausible. Accordingly, a second mechanism has been suggested in which integration does not (necessarily) rely on conjunction detectors. The idea is that integration comprises of *synchronizing* the firing patterns of feature-coding neurons, in such a way that the neurons coding for features of the same object act as a unit (Engel & Singer, 2001; Raffone & Wolters, 2001; von der Malsburg, 1999).

Even though these two mechanisms are commonly treated as mutually exclusive alternatives, the benefits and costs they imply suggest that they both play a role in dealing with binding problems (Colzato, Raffone & Hommel, 2006; VanRullen, 2009 this issue). Synchronization-based integration has the advantage of being particularly flexible and parsimonious in terms of long-term memory structures but the disadvantage that even features that are very likely to cooccur would need to be bound anew every time they are encountered. Hence, this integration method would be economical in terms of cognitive structure but wasteful in terms of processing time. In contrast, convergence-based integration has the advantage of allowing for the fast and effortless registration of feature combinations of practically unlimited complexity but the disadvantage that (apart from possible genetically hardwired conjunction detectors) this registration presupposes extensive learning and some degree of separability of conjunctions (as conjunction with too much feature overlap would lead to the activation of too many conjunction detectors). Hence, this integration method would be economical in terms of processing time but wasteful in terms of structure. Given that our environment calls for both the recognition of highly reliable feature combinations (as with natural objects) and the

processing of highly arbitrary combinations (as commonly used in psychological experiments), it makes sense to assume that both convergence-based and synchronization-based mechanisms are at work in generating human perception.

The present study was conceived of with this distinction in mind. As we will argue, there is behavioural evidence suggesting that at least two different mechanisms are at work in human visual feature integration, one that operates on any feature conjunction that is encountered (presumably based on synchronization) and another that selectively operates on frequent conjunctions that have been stored (presumably based on convergence). Before addressing the motivation and rationale of our study in more detail we will provide a brief overview of the theoretical background relevant for the study and the experimental paradigm we used.

OBJECT FILES

Most studies addressing feature binding focused on the visual modality. Some studies investigated whether there actually are binding problems in visual feature integration. For instance, Treisman and Schmidt (1982), and many others since then, demonstrated that creating attentionally demanding conditions results in an increasing numbers of incorrect bindings or “illusory conjunctions”—suggesting that feature integration is not a trivial task and raising doubt whether it is solely handled by convergence mechanisms (which have a hard time predicting such observations). Other studies have looked into whether people actually *do* bind features, even under circumstances that do not seem to require any binding. Particularly appropriate for that purpose turned out to be the preview paradigm developed by Kahneman, Treisman, and Gibbs (1992) and used by many others since then.

The simplest, stripped-down form of this paradigm is illustrated in Figure 1 (please ignore the R1 cue and R1 for the moment). The sequence of trial events comprises a nominally task-irrelevant prime or preview display (S1) followed by a probe display (S2). Let's assume in this example that S1 and S2 can either be a circle or triangle, can be red or green, and can be presented in the top or bottom position. The observer's task is simply to identify S2's shape as quickly as possible. Colour and position information are completely irrelevant to the task and can be safely ignored. Because S1 requires no response, it too can be ignored. Critically, S1 may consist of the same or a different shape as S2 and may appear in the same or a different colour and at the same or a different location. Analysing performance on S2 (i.e., R2) as a function of the repetition or alternation of shape, colour, and location (from S1 to S2) may or may not produce main effects, such as better performance if a particular feature repeats. These kinds of effects should

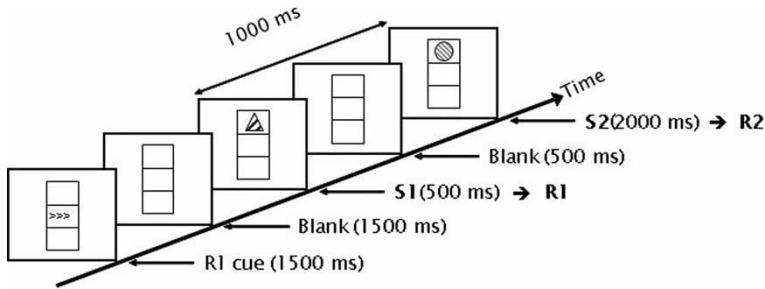


Figure 1. Sequence of events in the present experiment. Participants prepared a left- or right-hand response as indicated by a cue (R1 cue) and carried it out (R1) when the next stimulus (S1) appeared. Then they waited for the next stimulus (S2) and carried out a left- or right-hand response (R2) to its shape. S1 and S2 varied in shape, colour, and location, so that all three features could repeat or alternate. As R1 did not depend on S1 (but on the R1 cue), the response could repeat or alternate irrespective of the stimulus sequence.

reflect the priming of feature codes (e.g., leftover activation in the corresponding codes) and are thus unlikely to reflect binding processes (Kahneman et al., 1992).

Theoretically more interesting are *interactions* between repetition effects, because they indicate that the impact of repeating one feature depends on the identity of the other. Such interactions have been frequently observed: Repeating one feature produces better performance if other features are also repeated (e.g., Kahneman et al., 1992) but worse performance if these other features alternate (e.g., Hommel, 1998). To take our example, responding to a red circle (as S2) is easier following the presentation of another red circle than following a green circle (as S1), but more difficult following a red than a green triangle. Likewise, responding to a top circle is easier following a top than a bottom circle, but more difficult following a top than a bottom triangle. These observations suggest that registering the cooccurrence of two or more given features is sufficient to create some sort of binding between them, so that reencountering one of them tends to reactivate the whole binding in a pattern-completion-like process (Hommel, 2004). As a consequence, partial repetitions induce the retrieval of no longer valid feature codes, which disturbs current coding processes and induces code conflict. There is thus evidence that people do bind features—no matter how plausible one finds the available arguments for or against the logical necessity of feature binding. And they do so even under the most unlikely circumstances, that is, even if neither the bound features nor the object they refer to need to be attended or reported, even in the absence of any attentionally challenging display or task, and even when no more than 250–500 ms are available for creating such bindings (Hommel & Colzato, 2004).

Direct evidence for the retrieval of object files has been obtained in an fMRI study by Keizer et al. (2008). They presented subjects with preview (S1) and probe displays (S2) that both consisted of two blended pictures showing a face and a house. Either the face or the house moved in one of two possible directions, and subjects were to respond to the direction of S2 irrespective of which object moved. Most interesting were the conditions in which S1 showed a moving house and S2 a moving face: If the direction of motion in these two displays was the same more activation of house-related information in the parahippocampal place area was observed than if the motion differed. This suggests that the direction of motion was integrated with the object that moved, so that repeating the motion reactivated the representation of the object that had just accompanied this motion. Hence, binding features creates episodic cognitive structures that tend to be reactivated as a whole as soon as one of their ingredients matches the current input. This is indeed what underlies the original idea underlying Kahneman et al.'s (1992) *object file* concept: Bindings are functional in establishing object constancy by maintaining information about an object even in the absence of current sensory input and by relating this information to later reoccurrences of this object, even though these reoccurrences may only match part of the maintained information. This is why we can track objects over longer periods of occlusion and across changes in a number of visual features.

Interactions between repetition effects indicative of feature binding have been obtained for various features. Shape, colour, and location features have been shown to interact (Hommel, 1998, 2007; Hommel & Colzato, 2004) just as well as face, house, and motion information (Keizer, Colzato & Hommel, 2008)—suggesting that binding can span ventral and dorsal processing streams. An interesting observation in all these studies is that location does not seem to play a particularly dominant role. Some authors have claimed that object files can only be reactivated or reassessed if the current object matches the respective object file in terms of location (Kahneman et al., 1992; Mitroff & Alvarez, 2007). This would suggest that nonspatial matches are insufficient by themselves to retrieve a previous object file, which again implies that nonspatial features can only interact if this interaction is mediated by a location match. Even though it is clear that spatial location and spatial matches are important in multielement displays, simply because location is commonly crucial to track the identity of an object, spatially unmediated interactions between nonspatial features are possible (e.g., Colzato, Raffone, & Hommel, 2006; Hommel, 1998, 2007), which disconfirms approaches that rely on spatial correspondence as a retrieval cue. On the other hand, spatial location clearly plays a central role in the *encoding* of object files. Most studies on feature integration confound the sharing of spatial location with belongingness to the same perceptual object. Van Dam

and Hommel (2008) disentangled these factors by testing whether two given features appearing in the same location would be still integrated even if they obviously belonged to two different objects. Indeed, orientation and colour features were bound (i.e., orientation- and colour-repetition effects interacted) irrespective of whether they appeared as part of the same object or of different objects (e.g., one stationary and the other moving continuously, or a banana in a particular orientation overlaying an apple of a particular colour). In contrast, integration was markedly reduced when the two objects were separated in space (cf. Xu, 2006). Thus, spatial location is important for the encoding but not the retrieval or reactivation of object files.

MULTIPLE INTEGRATION MECHANISMS

The available evidence suggests that cooccurring visual features are more or less automatically bound into object files, that is, temporary links between, or pointers to the codes representing, the features of a perceived visual event. To take our example, and following Hommel (2004) and Colzato et al. (2006), this process can be captured by the cartoon model sketched in Figure 2a. Registering a red circle appearing in a top position would lead to the activation of corresponding codes in shape, colour, and location maps, and these codes are cross referenced by creating a temporary object file (symbolized by the folder).

Note that Figure 2 considers a further impact from the current attentional set, which is assumed to prime task-relevant feature dimensions (shape in our example). The reason to include such a top-down mechanism derives from a number of observations. On the one hand, the fact that bindings are created under the most unlikely conditions seems to suggest that feature integration is a highly automatic process. Indeed, systematic manipulations of the amount of attention directed to the to-be-integrated features and available for integration failed to modulate feature integration effects (Hommel, 2005, 2007), suggesting that the encoding of event files is a spontaneous process (cf. Logan, 1988). On the other hand, the retrieval of event files turned out to be rather highly controllable. One indication for that is that features varying on task-relevant feature dimensions are more likely to be involved in interactions with other features. For instance, having participants respond to the shape of S2 yields particularly strong binding-related effects involving shape repetition while having them respond to colour yields particularly strong effects involving colour repetition (e.g., Hommel, 1998). Even trial-to-trial shifts between shape- and colour-relevant versions of the task induce stronger binding-related effects for the currently task-relevant feature dimension, suggesting that attentional set has an immediate impact (Hommel, Memelink, Zmigrod, & Colzato, 2008). As it

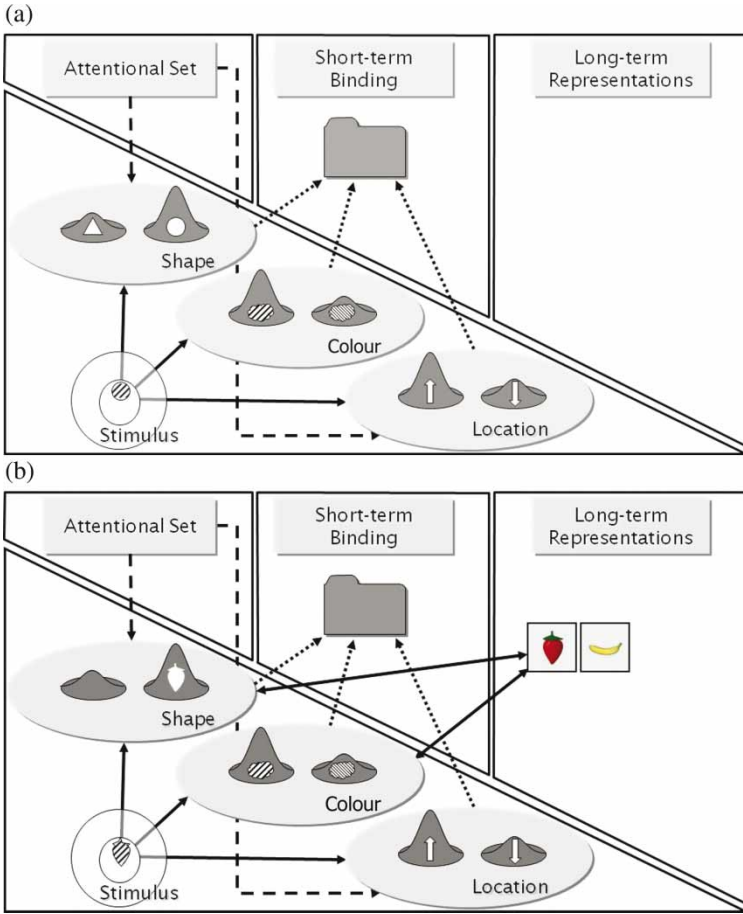


Figure 2. Cartoon model of feature binding and binding retrieval. (a) The coding of arbitrary feature conjunctions (adapted from Colzato et al., 2006). Colours are indicated by fill patterns. Shape is directly task relevant (by having it signalling responses) and location is indirectly task relevant (by defining the responses in terms of spatial locations), so that the shape and location dimensions are primed by the attentional set. Features on primed dimensions are assumed to be more likely integrated with other features and/or retrieved by stimuli that feature-overlap with the respective binding. (b) The coding of highly familiar feature conjunctions or real objects represented in long-term memory. Stimuli that match long-term representations to a sufficient degree activate these representations, which again induce top-down priming of the stimulus features that are coded on object-defining dimensions (shape and colour in the example). Top-down priming may work by increasing the gain of the respective feature dimension, which multiplies the stimulus-induced activation of this dimension's feature codes. Note that short-term bindings and long-term representations relate differently to individual feature codes: Whereas short-term bindings are considered to “point” to, and thus remain linked with the individual codes, long-term representations only respond to the presence of features or feature combinations. Likewise, short-term bindings can reactivate individual feature codes upon retrieval of a code that has been bound with them, whereas the top-down effect of long-term representations is restricted to priming feature dimensions but not individual codes.

does not matter whether the set is established before or after the hypothetical binding process (i.e., before after S1 presentation), it makes sense to assume that the impact of the attentional set selectively targets object file *retrieval* but not encoding (Hommel et al., 2008).

In Figure 2a we have considered two types of impact of the current attentional set. One is rather obvious: Having participants to respond to the shape of stimuli makes the shape dimension task relevant, so that it should receive top-down priming. However, stimuli are not the only events in reaction time tasks; participants also need to prepare and carry out particular actions. Given that actions can be assumed to be cognitively represented in terms of their perceptual effects (reflecting the ideomotor principle; see Hommel, Müsseler, Aschersleben, & Prinz, 2001; James, 1890; Lotze, 1852), preparing and controlling a particular set of actions entails attending the perceptual dimensions on which the actions are defined (Adam, Hommel, & Umiltà, 2003; Fagioli, Hommel, & Schubotz, 2007). If, as in our example, responses are defined by their spatial location (left vs. right keypress), this introduces task relevance of location in general. If we assume that the task relevance of a feature dimension leads to the priming of all feature values defined on it (Fagioli et al., 2007; Found & Müller, 1996; Hommel, 2007), making left and right *response* locations relevant should lead to the priming of any location information (i.e., whether it refers to a stimulus event or a response)—just as indicated in Figure 2a. Accordingly, the impact of (repetitions of) *stimulus* location increases as a consequence of choosing spatial responses. Indeed, Hommel (2007) could provide evidence that repetitions of stimulus location strongly affect binding-related effects with spatially defined response sets but not with nonspatial responses.

The model shown in Figure 2a suffices to account for the basic findings from most studies using the design introduced by Kahneman et al. (1992), and the binding process it implies reflects more characteristics of synchronization-based binding than of convergence-based integration. For one, it is difficult to see how convergence detectors might produce partial-repetition costs. Clearly, activating the same detector twice, as with the complete repetition of a particular shape-colour conjunction, say, should speed up performance and it commonly does. But consider a partial repetition (of either the shape or the colour) and a nonrepetition. Partial repetitions may be thought to also activate the same detector twice, though to a lower degree, or to fail activating the same detector. In the first case one would expect that performance for partial repetitions falls between complete repetitions and nonrepetitions; whereas in the second case one would expect that performance on partial repetitions and nonrepetitions is equally worse than on complete repetitions. As already mentioned, however, the standard finding is that performance on complete repetitions and nonrepetitions is equally good and better than on partial repetitions (e.g., Hommel, 1998). Closer

consideration of the possible noise and competition between alternative conjunction detectors helps a bit. For instance, one may assume that encountering one combination of two possible colours and two possible shapes (a red circle, say) leads to the activation of at least three conjunction detectors: A strong activation of the target detector ($\langle \text{red} \rangle + \langle \text{circle} \rangle$) and some milder activation of feature-sharing detectors (say, $\langle \text{red} \rangle + \langle \text{square} \rangle$ and $\langle \text{green} \rangle + \langle \text{circle} \rangle$). If the target repeats, all three detectors would be reactivated; this would lead to a particularly strong activation of the target detector, which now dominates the feature-sharing detectors even more. More concretely, if we count three activation units per target-induced activation and one unit per activation through feature overlap, this would mean that the target detector as an activation level of 6 as compared to an activation level of 2 for each of the two main competitors. It is easy to see that this approach correctly predicts worse performance with partial repetitions: If a red circle is followed by a green circle, say, this would first activate the same three detectors ($\langle \text{red} \rangle + \langle \text{circle} \rangle = 3$, $\langle \text{red} \rangle + \langle \text{square} \rangle = 1$, and $\langle \text{green} \rangle + \langle \text{circle} \rangle = 1$) as in the previous example and then the target detector ($\langle \text{green} \rangle + \langle \text{circle} \rangle = 3$) and the feature-overlapping detectors ($\langle \text{green} \rangle + \langle \text{square} \rangle = 1$ and $\langle \text{red} \rangle + \langle \text{circle} \rangle = 1$). Now the activation level of the target detector would be 4, just as much as its main competitor ($\langle \text{red} \rangle + \langle \text{circle} \rangle$), a situation that should result in impaired performance. The approach can also account for the observation that nonrepetitions produce better performance than partial repetitions. If a red circle is followed by a green square, this would activate ($\langle \text{red} \rangle + \langle \text{circle} \rangle = 3$, $\langle \text{red} \rangle + \langle \text{square} \rangle = 1$, and $\langle \text{green} \rangle + \langle \text{circle} \rangle = 1$) followed by ($\langle \text{green} \rangle + \langle \text{square} \rangle = 3$) and ($\langle \text{green} \rangle + \langle \text{circle} \rangle = 1$ and $\langle \text{red} \rangle + \langle \text{square} \rangle = 1$). The target activation would now add up to 3 and face activation levels of 2 in each of the two main competitors. Even though the target is now exposed to stronger overall competition than with partial repetitions (3:4 as compared to 4:4), the target would be in the position to outcompete each of the two competitors separately. Hence, choosing the right parameters, one may end up with a model that can account for better performance under nonrepetition than partial repetition. But again, it is difficult to see how it can account for *equal* performance under complete repetitions and nonrepetitions.

These difficulties suggest looking for alternative or at least additional mechanisms that are able to integrate features. Particularly promising with regard to the available findings seems the assumption that features are bound by an ad hoc binding mechanism, such as synchronizing the firing patterns of feature codes. If such a mechanism has just bound, say, the feature codes $\langle \text{red} \rangle$ and $\langle \text{circle} \rangle$, and if this binding has been maintained or stored, repeating one but not the other feature (as with a red square) could retrieve this binding (and thus reactivate the codes $\langle \text{red} \rangle$ and

<circle>), which would result in coding conflict between <circle> and <square> codes (Hommel, 2004). Coding conflict would only occur with partial repetitions but not with complete repetitions (where there is no conflict) or with alternations (where there is no retrieval). Another argument for a role of synchronization derives from the observation that manipulations targeting the muscarinic-cholinergic transmitter system affect both visual binding and synchronization in the visual cortex. Muscarinic-cholinergic agonists and antagonists have been demonstrated to respectively facilitate and impair neural synchrony in the gamma band ($\sim 30\text{--}70$ Hz) in the visual cortex of the cat (Rodriguez-Bermudez, Kallenbach, Singer, & Munk, 2004) and muscarinic-cholinergic antagonists were found to interfere with feature binding in the rat (Botly & de Rosa, 2007). Consistent with that, binding-type interactions between repetitions of visual features in humans are boosted by caffeine (a muscarinic-cholinergic agonist) and reduced by alcohol (a muscarinic-cholinergic antagonist), but unaffected by nicotine (a nicotinic-cholinergic agonist that does not affect muscarinic pathways; Colzato, Erasmus, & Hommel, 2004; Colzato, Fagioli, Erasmus, & Hommel, 2005). These parallels are consistent with the assumption that visual feature binding is mediated by neural synchronization processes that are driven by muscarinic-cholinergic neurotransmitters (Colzato et al., 2005).

Recent observations however suggest that the model sketched in Figure 2a is incomplete in important ways. In a series of experiments, Colzato et al. (2006) investigated the relationship between binding and longer term learning. The basic idea was that learning particular feature conjunctions may change the way the features they entail are bound, by either increasing the strength of the binding (because it would find increasing support by long-term associations) or by eliminating binding effects (because online binding would be no longer necessary). Surprisingly, however, even though more frequent feature combinations facilitated performance as such, binding-related effects were not at all affected by learning. This was true for highly frequent arbitrary conjunctions of features, such as orientation and colour, and for real objects, like red strawberries and yellow bananas. Even though performance was better if strawberries appeared in red and bananas in yellow, there was no indication that, say, strawberries are more strongly (or weakly) bound to red than they are to yellow, pink, or purple. These findings suggest that binding and learning are less intimately related than one may think. Interestingly, however, a comparison between the different experiments of Colzato et al. suggested that real objects created larger partial-repetition costs (i.e., binding-related effects) than arbitrary combinations of simple features did, which may point to a contribution from long-term memory.

To account for this pattern of results, Colzato et al. (2006) suggested that feature integration may proceed via two routes, the ad hoc binding of

cooccurring features (presumably mediated by synchronization processes) and the registration of previously acquired conjunctions by conjunction detectors stored in long-term memory (presumably using convergence mechanisms). This brings into play long-term memory representations the way we sketched in Figure 2b. Whereas ad hoc integration takes place as described in Figure 2a, overlearning feature conjunctions are thought to establish a conjunction detector in long-term memory. These detectors may be of any complexity and thus function as object representations or cardinal cells in the sense of Barlow (1972). However, establishing a new detector makes sense only, so we suggest at least, under two conditions. First, the conjunction it can detect needs to be *significant* in the sense of reliably indicating a particular stimulus event and, second, the conjunction needs to be *diagnostic* in the sense that it should be functional in discriminating the given stimulus event from other events. The rationale of this reasoning is that devoting (presumably limited) cognitive structure to a task that in principle could also be solved by ad hoc binding presupposes some surplus functionality, which would not be given if a new conjunction detector would be unable to reliably detect the stimulus it stands for or discriminate it from alternative stimuli.

We further assume that, whenever a particular stimulus activates such a conjunction detector, the detector will provide top-down support by facilitating the processing of all the features belonging to the stimulus (Colzato et al., 2006)—which may be achieved by priming the respective stimulus dimensions and thus multiplying any stimulus-induced activation of codes falling on them. This assumption is grounded in evidence coming from several lines of research showing that it is easier to attend multiple features of the same object (e.g., Baylis & Driver, 1993; Duncan, 1984) and more difficult to ignore distractors if they are part of the same object (e.g., Baylis & Driver, 1992; Hommel, 1995; Kahneman & Henik, 1981). This implies that processing one feature of an object automatically opens the attentional gate to other feature dimensions of this object, whether this is useful or not. In the example shown in Figure 2b, this kind of top-down priming would facilitate the processing of shape and colour information, and of any other visual feature belonging to a dimension that defines the stimulus object. Given that stimulus location is not an object-defining feature, location information would not benefit from this top-down priming, however.

Two parallel mechanisms of feature integration could account for the observations of Colzato et al. (2006) in the following way. With arbitrary, not highly overlearned feature combinations that do not signify a unique object (as geometric shapes are commonly not related to or correlated with particular colours), the situation would be as depicted in Figure 2a: Shape coding would be primed, due to the task relevance of shape, but colour coding would not (stimulus location was not varied in that study).

Accordingly, even if shape–colour conjunctions would be automatically integrated, colour-induced retrieval would be weak at best and the corresponding effects would be modest and fragile. Indeed, Colzato et al. found only small effects reflecting shape–colour binding and even these effect tended to disappear with increasing practice (presumably due to increased focusing on the relevant shape information). Real objects with which participants are familiar would be more likely to have led to the creation of reliable and discriminative conjunction detectors or object representations in long-term memory. As depicted in Figure 2b, this would lead to a match between stimuli and the long-term representation and thus induce top-down facilitation of the object features, including both shape and colour. Accordingly, it would matter less that colour is actually not relevant for the task, implying stronger and more stable shape–colour binding effects—exactly what Colzato et al. observed.

AIM OF STUDY

The present experiment was set up to test the post hoc considerations of Colzato et al. (2006) in a more systematic fashion. We used a similar task (as sketched in Figure 1) but compared real objects and arbitrary feature conjunctions that varied on three dimensions (shape, colour, and stimulus location) in a within-subjects design. Shape was directly relevant for the task as participants were to discriminate and respond to the shape of S2. Even though stimulus location varied randomly and could safely be ignored, using spatially defined *responses* (left vs. right keypress) made location indirectly task relevant. Colour was entirely irrelevant. We expected the standard interactions between feature-repetition effects indicative of feature binding but were particularly interested in testing three more specific hypotheses.

First, we expected that interactions between shape and colour repetition (indicating shape–colour binding) would be more pronounced, and perhaps even restricted to, real objects. As explained already, real objects are likely to match representations stored in long-term memory, which should induce top-down priming of all object-related features. As shape is primed by task relevance anyway, it would be colour coding that benefits from this priming process, so that colour codes would interact more strongly with shape codes in the real-object condition.

Second, we expected that the difference between arbitrary feature combinations and real objects would not affect stimulus-location coding and, thus, not mediate location-related interactions. As explained earlier, long-term representations are unlikely to contain information about the location of a given object in space, as location is not an object-defining

attribute. Accordingly, location coding would not receive or benefit from top-down priming.

Third, we expected that the hypothetical real-object effect would be restricted to *stimulus-related* feature integration (and/or retrieval). Previous studies have shown that feature binding as such is not restricted to stimulus processing but operates across perception and action. Hommel (1998) has extended the classical preview design to include response repetitions by having participants to respond to the first stimulus (S1) with a previously cued and already prepared response (R1; see Figure 1). As in this design R1 does not correlate with the features of S1, stimulus features and responses can vary independently, so that stimulus–feature repetition and response repetition can be orthogonally manipulated (so to avoid the acquisition of stimulus–response associations). If this is done, the same type of crossover interaction as with stimulus–feature repetitions can be observed: Repeating a stimulus feature facilitates performance if the response also repeats but impairs performance if the response alternates (Hommel, 1998). Again, it seems that the mere single cooccurrence of a stimulus attribute and a response is sufficient to create a binding between their codes, so that repeating either the attribute or the response is sufficient to reactivate both or all members of this binding. Indeed, repeating some of the stimulus attributes induces a tendency to repeat the response as well in a free-choice reaction task (Hommel, 2007). However, visuomotor integration clearly differs from the process responsible for the integration of visual features. Not only do the two integration processes operate at a different point in time (visual binding seems to be stimulus locked, whereas visuomotor binding seems to be response locked; Hommel, 2005) but they are also driven by different neurotransmitter systems: Whereas muscarinic-cholinergic manipulations affect visual but not visuomotor binding (Colzato et al., 2004, 2005), manipulations targeting dopaminergic pathways affect visuomotor but not visual binding (Colzato & Hommel, 2008; Colzato, Kool, & Hommel, 2008; Colzato, van Wouwe & Hommel, 2007a, 2007b). Moreover, if it is true that establishing detectors for highly frequent, unique, and reliable feature conjunctions serves the purpose of facilitating object perception, it makes sense to assume that the impact of such detectors are restricted to perceptual processes—visual integration that is. Accordingly, we expected the standard interactions between visual-feature repetition and action repetition (Hommel, 1998) but no mediation of these effects by the arbitrary-conjunction vs. object manipulation.

METHOD

Thirty students of the Leiden University served as subjects for partial fulfilment of course credit or a financial reward. All reported having normal or corrected-to-normal vision, and were not familiar with the purpose of the experiment.

The experiment was controlled the Experimental Run-Time System (ERTS™) running on a PC attached to a 17-inch monitor. Participants faced three grey square outlines, vertically arranged, as illustrated in Figure 1. From a viewing distance of about 60 cm, each of these frames measured $2.6^\circ \times 3.1^\circ$. A banana ($0.3^\circ \times 0.6^\circ$), a strawberry ($0.5^\circ \times 0.6^\circ$), a triangle ($0.3^\circ \times 0.6^\circ$), and a circle ($0.5^\circ \times 0.6^\circ$) served as S1 and S2 alternatives, which were presented in blue or (close-to-magenta, purplish) pink (to avoid any preexperimental object–colour associations)¹ in the top or bottom frame. The stimuli were taken from Experiment 4 of Colzato et al. (2006; see p. 711 for bitmaps) and presented in the same colours (parameters were red = 0, green = 0, blue = 255, hue = 160, saturation = 240, and luminance = 120, for blue and red = 255, green = 0, blue = 255, hue = 200, saturation = 240, and luminance = 120, for pink). Response cues were also presented in the middle frame (see Figure 1), with rows of three left- or right-pointing arrows indicating a left and right keypress, respectively. Responses to S1 and to S2 were made by pressing the left or right shift-key of the computer keyboard with the corresponding index finger.

The experiment consisted of two sessions of 35 min, one with real objects (banana and strawberry) and one with arbitrary feature conjunctions (triangle and circle). In both sessions participants carried out two responses per trial, a previously cued simple response (R1) and a binary-choice

¹ More specifically, our idea was to get the hypothesized object representations in long-term memory involved—which required the use of stimuli that were likely to have memory representations—without letting them do the integration job on their own (i.e., without the need for ad hoc feature binding)—which required the use of feature combinations that were unlikely to be covered by these representations. Following Colzato et al. (2006), we thus used shapes of real objects (which should suffice to activate the memory representations) but presented them in colours that were unlikely to be part of the memory representation—using two colours that according to the findings of Colzato et al. are not associated with either of the two object shapes. Theoretically speaking, we expected that this manipulation would activate object representations and the corresponding conjunction detectors but still require ad hoc binding of the uncommon shape–colour conjunction. The former was considered to provide top-down priming of the latter with real objects but not with arbitrary feature conjunctions. As an example, facing a banana should activate a banana-related conjunction detector, which would lead to top-down priming of *all* identity-relevant features belonging to the *present* banana (i.e., to both the familiar shape and the in this case unfamiliar colour). This would prime the colour and increase the likelihood that it is being integrated. As the geometric shapes were not considered to have conjunction detectors linking them to particular colours, no top-down priming should occur for these shapes and present their colours.

response (R2) to the shape of the second of two target stimuli (S1 and S2; see Figure 1). They first prepared a left- or right-hand response as indicated by a cue (R1 cue) and carried it out (R1) upon presentation of the next stimulus (S1). S1 thus merely triggered the already prepared response; its features were entirely irrelevant and uncorrelated with the response. Then participants awaited the next stimulus (S2) and carried out a left- or right-hand response (R2) to its shape. Participants were informed that there would be no systematic relationship between S1 and R1, or between S1 and S2, and they were encouraged to respond to the onset of S1 only, disregarding the stimulus' attributes. The mappings of stimuli to response keys (S2 → R2) and the order of sessions were balanced across participants. The sequence of events in each trial is shown in Figure 1. The experiment was composed of 512 trials resulting from a factorial combination of the two possible shapes, colours, and locations of S2, the stimulus-type (real objects vs. arbitrary feature conjunctions) and the repetition vs. alternation of shape, colour, stimulus location, and the response, and three replications per condition.

RESULTS AND DISCUSSION

After excluding trials with missing (> 1500 ms) or anticipatory responses (< 200 ms), mean reaction times (RTs) and proportions of errors for R2 were analysed (see Table 1 for means and Table 2 for ANOVA terms). ANOVAs were run with stimulus type (real objects vs. arbitrary feature conjunctions), the repetition versus alternation of stimulus shape, colour, and location (S1 → S2), and of the response (R1 → R2) as within-participant factors.

In RTs, the main effect of stimulus type indicated that subjects reacted faster to the arbitrary feature conjunctions than to real objects and the stimulus-location repetition costs in RTs and errors reflect “inhibition of return”—the common observation that attending to an irrelevant stimulus impairs later responses to relevant stimuli appearing in the same location (Posner & Cohen, 1984). More interesting for present purposes were the interactions. First, shape repetition interacted with stimulus location (in RTs) and with colour (in RTs and errors)—repeating one object feature but not the other feature impaired performance as compared to complete repetitions and alternations. Importantly, only the shape-by-colour interaction was modified by stimulus type, thus producing a three-way interaction. As suggested by Figure 3, the shape-by-colour interaction was considerably more pronounced with real objects, $F(1, 29) = 28.14$, $p = .0001$, than with arbitrary feature conjunctions, where the interaction was not reliable, $F(1, 29) = 1.39$, $p = .247$. We checked whether the stronger interaction with real objects might be due to the higher RT level in this condition. However, the

TABLE 1

Means of mean reaction times and standard deviations (*SD*) for responses to Stimulus 2 (RT, in ms) and percentages of errors on R2 (PE), as a function of stimulus type (real objects vs. arbitrary feature conjunctions), the match between Response 1 and Response 2, and the feature match between Stimulus 1 and Stimulus 2

Match	Response							
	Arbitrary feature conjunctions				Real objects			
	Repeated		Alternated		Repeated		Alternated	
	RT (<i>SD</i>)	PE (<i>SD</i>)	RT (<i>SD</i>)	PE (<i>SD</i>)	RT (<i>SD</i>)	PE (<i>SD</i>)	RT (<i>SD</i>)	PE (<i>SD</i>)
Neither	482 (19)	10.6 (2.4)	474 (21)	2.3 (1.9)	511 (20)	10.6 (1.9)	482 (22)	2.1 (1.0)
Shape (S)	499 (25)	7.9 (1.7)	492 (21)	10.2 (2.0)	510 (20)	7.7 (1.2)	515 (21)	7.9 (1.7)
Location (L)	522 (27)	9.6 (2.0)	495 (21)	2.7 (0.1)	532 (23)	7.7 (1.9)	509 (17)	5.8 (1.3)
Colour (C)	497 (26)	14.2 (2.9)	463 (21)	2.7 (0.7)	512 (24)	10.8 (2.2)	483 (19)	4.3 (1.2)
S × L	475 (19)	3.9 (1.3)	499 (17)	13.7 (1.9)	497 (20)	4.8 (0.8)	523 (16)	14.6 (2.2)
C × L	512 (24)	10.4 (2.1)	498 (23)	6.9 (1.1)	549 (27)	10.2 (2.5)	528 (22)	7.1 (2.0)
S × C	494 (27)	5.4 (1.4)	478 (22)	8.1 (1.6)	495 (22)	5.2 (0.8)	510 (20)	5.0 (1.1)
S × L × C	462 (20)	3.7 (1.1)	497 (20)	14.6 (3.0)	477 (19)	2.5 (0.9)	499 (20)	12.7 (2.6)

outcome of two analyses speaks against this possibility. First, we median-split participants by their mean RT in the real-object condition and reran the ANOVA with level as additional between-participant variable. Whereas the three-way interaction was still reliable, $F(1, 28) = 4.80, p = .04$, there was no hint for any mediation by level, $F(1, 28) < 1$. Second, we computed, for each participant, the increase in shape-colour effect size and the increase in RT from the arbitrary-feature-conjunction condition to the real-object condition.² Given that the two measures were uncorrelated, $r = .06, p > .7$, it seems safe to conclude that a higher RT level as such does not increase the shape-by-colour interaction. The observation that the interaction is mediated by the type of stimulus fully supports our expectation that real objects provide top-down priming for colour coding that compensates for the lack of task relevance. Accordingly, colour did not interact with any other feature dimensions in the case of arbitrary conjunctions but it did interact with shape, the other object-specific feature dimension, in the case of real objects.

A second cluster of interactions in both RTs and error rates involved response repetition. It interacted with shape repetition and with stimulus-location repetition, and was involved in a three-way interaction with shape

² Effect sizes were computed by subtracting the mean RT for complete repetitions and alternations from the mean RT for partial repetitions (i.e., shape repetition and colour alternation or shape alternation and colour repetition). Note that this amounts to the interaction term corrected for possible main effects.

TABLE 2
 Results of analysis of variance on mean reaction time of correct responses (RT) and percentage of errors (PE)

Effect	RT_{R2}		PE_{R2}	
	MSE	F	MSE	F
Stimulus type (T)	16069.44	5.04*	52.98	0.07
Shape (S)	6254.84	2.39	112.02	0.84
Colour (C)	6909.71	0.58	61.11	0.43
Location (L)	3292.18	8.92**	38.78	6.06*
Response (R)	55221.34	1.32	223.37	0.45
T × S	1362.66	0.50	45.02	0.01
T × C	1991.67	0.10	62.18	0.09
T × L	1902.42	0.05	48.32	0.05
S × C	1277.82	13.62***	47.95	16.16**
T × S × C	916.60	4.97*	31.46	0.25
C × L	1297.17	0.02	81.96	0.87
T × C × L	1523.76	0.13	61.09	1.41
S × L	8461.69	10.01**	65.27	1.56
T × S × L	1800.15	0.36	36.28	0.54
S × C × L	4844.92	0.28	53.57	0.30
T × C × S × L	2747.68	2.06	53.76	1.74
T × R	2574.38	0.08	80.22	0.24
S × R	2005.57	36.98***	169.77	44.73***
T × S × R	1665.66	1.64	54.28	2.74
C × R	1942.08	0.02	47.21	0.22
T × C × R	1374.84	0.37	30.53	0.01
S × C × R	1672.61	0.13	45.61	0.60
T × S × C × R	1450.68	0.05	35.13	0.12
L × R	1281.34	12.20**	132.17	26.25***
T × L × R	1038.80	2.21	49.11	0.48
C × L × R	1715.44	1.36	59.88	1.69
T × C × L × R	3157.48	1.61	48.81	1.76
S × L × R	1086.53	7.92**	44.76	10.21**
T × S × L × R	1759.39	2.56	44.42	0.29
C × S × L × R	2297.17	0.37	55.09	0.29
T × C × S × L × R	862.35	0.01	50.56	1.85

* $p < .05$, ** $p < .01$, *** $p < .001$; $df = 1, 29$ for all effects.

and stimulus location. The latter indicated that location only interacted with the response if shape was repeated, $F(1, 29) = 31.92$, $p < .001$ (RTs), and $F(1, 29) = 24.89$, $p < .001$ (errors), but not if shape alternated, $ps > .4$. The reliable interactions all followed the standard form with better performance for shape repetitions and for stimulus-location repetitions if the response repeated but worse performance for shape repetitions and stimulus-location repetitions if the response alternated. Most interesting for present purposes, stimulus type and response repetition were not involved in any reliable

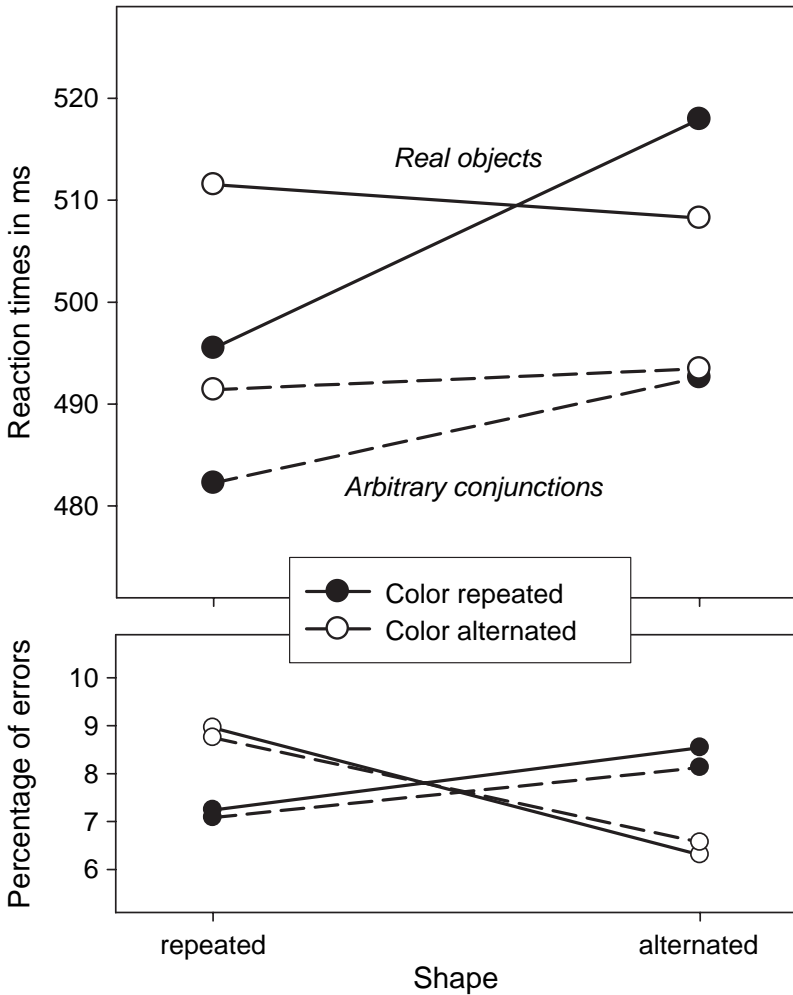


Figure 3. Mean reaction times and error percentages for R2 as a function of stimulus type and of repetition versus alternation of stimulus shape and colour.

interaction. This suggests that visual integration is influenced by stimulus type but visuomotor integration is not.

CONCLUSIONS

Taken altogether, our findings support the assumption of two different feature-integration mechanisms in visual perception (see also VanRullen, 2009 this issue). One mechanism seems to be agnostic about the familiarity

or possibility of particular combinations of features and integrates any feature that falls into a given temporal integration window (Akyürek, Toffanin, & Hommel, 2008). We speculate that this mechanism is mediated by, or relies on neural synchronization processes, as is suggested by the observation that both visual integration and synchronization in the visual cortex seem to be driven by the same neurotransmitter system. The other mechanism is sensitive to the familiarity with the stimulus and it seems to provide attentional top-down support for real, familiar objects. Recognizing such objects presupposes a stored detector of the underlying feature combinations, and we speculate that such detectors are created for frequent combinations only—even though we cannot rule out the possibility that the naturalness of the stimulus also plays a role. In any case, there are reasons to assume that feature integration can take place in more than one way and that the principles of integration-through-convergence and integration-through-synchronization do not exclude but complement each other.

More generally speaking, our findings provide support for the assumption that the retrieval of object files is cocontrolled by two types of top-down priming processes. Offline priming, as one may call the impact of the current attentional set, precedes the stimulus and reflects the task relevance of feature dimensions for selecting the stimulus and the response (see Figure 2a and b). This priming is offline in the sense that it can be established any time before a given stimulus or response event occurs. Online priming, as it may be characterized, can be induced by stimuli that have entries in long-term memory, such as familiar real objects. Sensory information coming from these stimuli is likely to access corresponding memory entries in a first fast forward sweep, followed by a recurrent top-down process refining and contextualizing the input (Lamme & Roelfsema, 2000). Whereas the first, bottom-up part of this scenario is likely to be rather nonselective, the recurrent wave will be shaped by the current attentional settings. Given that this wave follows the first contact between the sensory information and the memory content, the outcome of this contact will contribute as well. Accordingly, the eventual representation of the present stimulus and the degree to which this representation is permitted to reactivate available object files will thus be codetermined by the task set and the stimulus-induced memory activation—provided that the stimulus matched some memory content it could activate.

REFERENCES

- Adam, J., Hommel, B., & Umiltà, C. (2003). Preparing for perception and action I: The role of grouping in the response-cuing paradigm. *Cognitive Psychology*, *46*, 302–358.
- Akyürek, E. G., Toffanin, P., & Hommel, B. (2008). Adaptive control of event integration. *Journal of Experimental Psychology: Human Perception and Performance*, *34*, 569–577.

- Barlow, H. B. (1972). Single units and sensation: A neuron doctrine for perceptual psychology. *Perception, 1*, 371–394.
- Baylis, G. C., & Driver, J. (1992). Visual parsing and response competition: The effect of grouping factors. *Perception and Psychophysics, 51*, 145–162.
- Baylis, G. C., & Driver, J. (1993). Visual attention and objects: Evidence for hierarchical coding of location. *Journal of Experimental Psychology: Human Perception and Performance, 19*, 451–470.
- Botly, L. C. P., & de Rosa, E. (2007). Cholinergic influences on feature binding. *Behavioral Neuroscience, 121*, 264–276.
- Colzato, L. S., Erasmus, V., & Hommel, B. (2004). Moderate alcohol consumption in humans impairs feature binding in visual perception. *Neuroscience Letters, 360*, 103–105.
- Colzato, L. S., Fagioli, S., Erasmus, V., & Hommel, B. (2005). Caffeine, but not nicotine enhances visual feature binding. *European Journal of Neuroscience, 21*, 591–595.
- Colzato, L. S., & Hommel, B. (2008). Cannabis, cocaine, and visuomotor integration: Evidence for a role of dopamine D1 receptors in binding perception and action. *Neuropsychologia, 46*, 1570–1575.
- Colzato, L. S., Kool, W., & Hommel, B. (2008). Stress modulation of visuomotor binding. *Neuropsychologia, 46*, 1542–1548.
- Colzato, L. S., Raffone, A., & Hommel, B. (2006). What do we learn from binding features? Evidence for multilevel feature integration. *Journal of Experimental Psychology: Human Perception and Performance, 32*, 705–716.
- Colzato, L. S., van Wouwe, N. C., & Hommel, B. (2007a). Feature binding and affect: Emotional modulation of visuo-motor integration. *Neuropsychologia, 45*, 440–446.
- Colzato, L. S., van Wouwe, N. C., & Hommel, B. (2007b). Spontaneous eyeblink rate predicts the strength of visuomotor binding. *Neuropsychologia, 45*, 2387–2392.
- DeYoe, E. A., & van Essen, D. C. (1988). Concurrent processing streams in monkey visual cortex. *Trends in Neuroscience, 11*, 219–226.
- Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology: General, 113*, 501–517.
- Engel, A. K., & Singer, W. (2001). Temporal binding and the neural correlates of sensory awareness. *Trends in Cognitive Science, 5*, 16–25.
- Fagioli, S., Hommel, B., & Schubotz, R. I. (2007). Intentional control of attention: Action planning primes action-related stimulus dimensions. *Psychological Research, 71*, 22–29.
- Found, A., & Müller, H. J. (1996). Searching for unknown feature targets on more than one dimension: Investigating a “dimension weighting” account. *Perception and Psychophysics, 58*, 88–101.
- Hommel, B. (1995). Attentional scanning in the selection of central targets from multi-symbol strings. *Visual Cognition, 2*, 119–144.
- Hommel, B. (1998). Event files: Evidence for automatic integration of stimulus–response episodes. *Visual Cognition, 5*, 183–216.
- Hommel, B. (2004). Event files: Feature binding in and across perception and action. *Trends in Cognitive Sciences, 8*, 494–500.
- Hommel, B. (2005). How much attention does an event file need? *Journal of Experimental Psychology: Human Perception and Performance, 31*, 1067–1082.
- Hommel, B. (2007). Feature integration across perception and action: Event files affect response choice. *Psychological Research, 71*, 42–63.
- Hommel, B., & Colzato, L. S. (2004). Visual attention and the temporal dynamics of feature integration. *Visual Cognition, 11*, 483–521.
- Hommel, B., & Elsner, B. (in press). Acquisition, representation, and control of action. In E. Morsella, J. A. Bargh, & P. M. Gollwitzer (Eds.), *The psychology of action, Vol. 2*. Oxford, UK: Oxford University Press.

- Hommel, B., Memelink, J., Zmigrod, S., & Colzato, L. S. (2008). How information of relevant dimension control the creation and retrieval of feature–response binding. *Manuscript submitted for publication*.
- Hommel, B., Müsseler, J., Aschersleben, G., & Prinz, W. (2001). The theory of event coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, *24*, 849–878.
- James, W. (1890). *The principles of psychology*. New York: Dover Publications.
- Kahneman, D., & Henik, A. (1981). Perceptual organization and attention. In M. Kubovy & J. R. Pomerantz (Eds.), *Perceptual organization* (pp. 181–211). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.
- Kahneman, D., Treisman, A., & Gibbs, B. J. (1992). The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, *24*, 175–219.
- Keizer, A. W., Colzato, L. S., & Hommel, B. (2008). Integrating faces, houses, motion, and action: Spontaneous binding across ventral and dorsal processing streams. *Acta Psychologica*, *127*, 177–185.
- Keizer, A. W., Colzato, L. S., Theeuwisse, W., Nieuwenhuis, S., Rombouts, S. A. R. B., & Hommel, B. (2008). When moving faces activate the house area: An fMRI study of object file retrieval. *Manuscript submitted for publication*.
- Lamme, V. A., & Roelfsema, P. R. (2000). The distinct modes of vision offered by feedforward and recurrent processing. *Trends in Neuroscience*, *23*, 571–579.
- Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492–527.
- Lotze, R. H. (1852). *Medicinische Psychologie oder die Physiologie der Seele*. Leipzig, Germany: Weidmann'sche Buchhandlung.
- Mitroff, S. R., & Alvarez, G. A. (2007). Space and time, not surface features, underlie object persistence. *Psychonomic Bulletin and Review*, *14*, 1199–1204.
- Posner, M. I., & Cohen, Y. A. (1984). Components of visual orienting. In H. Bouma & D. G. Bouwhuis (Eds.), *Attention and performance X* (pp. 531–554). Hove, UK: Lawrence Erlbaum Associates Ltd.
- Raffone, A., & Wolters, G. (2001). A cortical mechanism for binding in visual working memory. *Journal of Cognitive Neuroscience*, *13*, 766–785.
- Rodriguez-Bermudez, R., Kallenbach, U., Singer, W., & Munk, M. H. (2004). Short- and long-term effects of cholinergic modulation on gamma oscillations and response synchronization in the visual cortex. *Journal of Neuroscience*, *24*, 10369–10378.
- Treisman, A. (1996). The binding problem. *Current Opinion in Neurobiology*, *6*, 171–178.
- Treisman, A., & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. *Cognitive Psychology*, *14*, 107–141.
- Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*, 97–136.
- Van Dam, W. O., & Hommel, B. (2008). How object-specific are object files? Evidence for integration by location. *Manuscript submitted for publication*.
- Van der Heijden, A. H. C. (1995). Modularity and attention. *Visual Cognition*, *2*, 269–302.
- VanRullen, R. (2009). Binding hardwired vs. on-demand feature conjunctions. *Visual Cognition*, *17*, 103–119.
- Von der Malsburg, C. (1999). The what and why of binding: The modeler's perspective. *Neuron*, *24*, 95–104.
- Xu, Y. (2006). Encoding objects in visual short-term memory: The roles of feature proximity and connectedness. *Perception and Psychophysics*, *68*, 815–828.

Selection of Robot Pre-Grasps using Box-Based Shape Approximation

Kai Hübner and Danica Kragic

Abstract—Grasping is a central issue of various robot applications, especially when unknown objects have to be manipulated by the system. In earlier work, we have shown the efficiency of 3D object shape approximation by box primitives for the purpose of grasping. A point cloud was approximated by box primitives [1]. In this paper, we present a continuation of these ideas and focus on the box representation itself. On the number of grasp hypotheses from box face normals, we apply heuristic selection integrating task, orientation and shape issues. Finally, an off-line trained neural network is applied to chose a final best hypothesis as the final grasp. We motivate how boxes as one of the simplest representations can be applied in a more sophisticated manner to generate task-dependent grasps.

I. INTRODUCTION

In a service robot scenario, robot grasping capabilities are necessary to actively execute tasks, interact with the environment and thereby reach versatile goals. These capabilities also include the generation of stable grasps to safely handle even objects unknown to a robot. In earlier work [1], we motivated the idea that the key to this ability is not primarily to select a grasp depending on the identification of a selected object, but rather on its shape. We presented an algorithm that efficiently wraps given 3D data points of an object into primitive box shapes by a fit-and-split algorithm based on Minimum Volume Bounding Boxes. Though box shapes are not able to approximate arbitrary data in a precise manner, it was shown that they give efficient clues for planning grasps on arbitrary objects or object parts. This seems reasonable, since it should not be necessary to find the most stable grasp, but sufficient to find one of those that are stable. Additionally, the part-describing box concept allows for grasp semantics mapped to boxes in the set, e.g. “*approach the biggest part to stably move the object*” or “*approach the smallest part to show a most unoccluded object to a viewer.*” The description of an object by a shape-based part representation, which is claimed to be necessary for this kind of task-dependent grasping, is thereby made available, and also needed as a criterion what grasp is the “best” in terms of a given task.

In this context, we present our novel approach for connecting shape, boxes, tasks and grasping in this paper. We briefly introduce our basic work as also other related work in Section II. While we refer to [1] for the description of the box decomposition algorithm, we focus on taking advantage of the box representation. We develop a sequence of steps, including heuristics and learning of grasp qualities to select

one final, task-dependent grasp for an object. We will discuss the simple ideas that are used to reach this goal in Section III. Section IV practically shows an experiment, where we connect to 3D data from a real, though convenient scene for the first time. We finally conclude our work in Section V.

II. RELATED WORK

When talking about a robot grasping unknown objects, one has to think about a representation that not only eases grasping, but which can also be efficiently delivered from the sensor data. Though there is interesting work on producing grasp hypotheses by visual features from 2D images only, e.g. [2], most techniques rely on 3D data. 3D data, which in its simplest form may be a set of 3D points belonging to an object’s surface, can be produced by several kinds of sensors and techniques, e.g. distance imaging cameras, laser scanners or stereo camera systems. Since the last solution is cheap, easy to integrate and close to the human sensory system, a multitude of concepts in the area use 3D point cloud data from stereo disparity. These point clouds are usually afflicted with sensor noise and uncertainties, which has to be taken into account for precise shape approximation of such data. In [1], we have referenced and stated our claim that precise shape approximation, e.g. using superquadrics, might not be necessary for extracting grasp hypotheses. The work of Lopez-Damian *et al.* [3], [4] is related to ours in terms of object decomposition and grasping. Additionally, they propose a grasp planner to find a stable grasp. However, their concept uses polygonal structures instead of 3D points. Though one could produce polygonal surfaces from 3D point data, for example by the Power Crust algorithm [5], this introduces another step causing additional effort both in processing time and noise handling. In this paper, we have also used the Power Crust, but only to visualize the 3D data.

It has to be mentioned that our approach is not explicitly handling contact-level grasp planning. A grounded theory on stable contact-level grasps has been developed in the literature. Conclusions of the ideas and outcome can be found in [6], [7]. In this theory of grasp planning, finger contact locations, forces and grasp wrench spaces can be simulated. Different criteria can be defined to rate grasp configurations, e.g. force closure, dexterity, equilibrium, stability and dynamic behavior [6]. However, the dependency on a-priori known or dense and detailed object models is apparent. Miller *et al.* [8] therefore proposed grasp planning on simple shape primitives, like spheres, cylinders and cones, clearly demanding a pre-classification of object shape. Dependent on the primitive shape, one can test several grasp configurations on this shape. This work was continued by Goldfeder *et*

The authors are with the KTH – Royal Institute of Technology, Stockholm, Sweden, as members of the Computer Vision & Active Perception Lab., Centre for Autonomous Systems, www home page: <http://www.csc.kth.se/cvap>, e-mail addresses: {khuebner, danik}@kth.se.

al. [9], using more sophisticated shape primitives, known as superquadrics.

In our work, we also work with shape primitives. We chose the box shape as one of the most simple ones and integrate an efficient bounding box algorithm for 3D point data [10]. However, while the classical contact-level solution includes a merge of both transport (leading the hand to the grasp position) and grip (closing the fingers to perform the grasp), we see a benefit in loosely decoupling these two components. The psychophysical shortcomings of completely decoupling the grip from the transport component have been discussed in [11], even if this is described as the classical approach. It is also hardly questioned that the transport component refers to extrinsic object properties only (e.g. position, orientation) while the grip component depends on intrinsic properties (e.g. size, shape, weight). Derbyshire *et al.* [12] even motivate action to be an intrinsic property.

The work presented here does neither separate nor combine these two components. It is more a connecting module inbetween them. First, the transport component is just seen as a predecessor. It would demand grasp planning and collision detection in a definition of successful robot hand transport, being a research topic for itself. However, the final location of a grasp is also clearly dependent on the task at hand, making the task another extrinsic property.

Second, the grip component is a successor of our grasp hypotheses generation. The final grip is not handled in a comparable way to classical contact-level grasp planning, as this connects directly to all perceptually sensed intrinsic properties. Thus, we classify our idea as a pre-grip component that is both dependent on selected extrinsic (orientation, task) and intrinsic (size, shape) properties (see bold in Tab. I). We see precise shape, weight or surface texture properties as being part of an adjacent fine-controller based on tactile feedback and corrective movements, like included in [13].

III. FEATURES OF THE BOX REPRESENTATION

The result of our box decomposition technique is the following: given a set of 3D points, we can find a compact box set $\mathcal{B} = \{B_1, \dots, B_n\}$ that encloses the points and thereby offers a primitive shape approximation. For each box B_i in the set, we focus on its six rectangular faces $\{F_{(i,1)}, \dots, F_{(i,6)}\}$. In [1], each face spawned up to four grasp hypotheses by using the face normal as approach vector and the four edges as orientation vectors, using a pre-defined grasp. Fig. 1 shows some of the models that were used, a model of a 5-finger hand, as also an exemplary box decomposition of the duck model. Finally, we showed that even if we drastically reduce the grasp hypotheses,

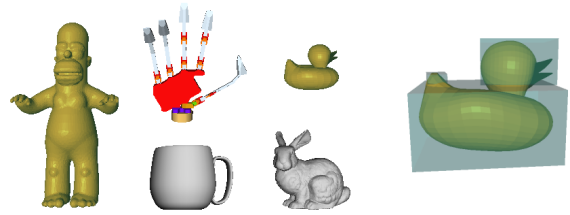


Fig. 1. Left: Some objects and a robot hand model, simulated in Graspt! [14]. Right: A result of the box approximation for the duck model [1].

this concept does not significantly reduce the grasp quality, but opens up new possibilities like task-oriented grasping or object part description. We will now present some of these issues, which have been integrated in a grasp selection mechanism, starting with task-dependencies.

A. Task Dependencies

Task dependency of grasps is an important issue, which shows that “best” grasps do not have to be the most stable ones. Picking up a cup from the “open side” will be unsuitable for the task of filling the cup, as a very stable full-enclosing grasp (power-grasp) will be unsuitable for handing over or presenting the cup to someone. Application of such re-usability semantics by defined keep-out zones has been proposed in [15]. Object properties like hollowness are hard to detect for today’s systems, as also are high-level properties like *filled* or *empty*. Our box set method allows intuitive mapping of less complex actions to simple box properties.

Given a box set \mathcal{B} , one can easily compute criteria like the overall mass center (assuming uniformly distributed mass density), each volume and dimension of a box, or the relations between boxes. For example, one can define the *outermost* or *innermost*, the *largest* or *smallest*, the *top* or the *bottom*, etc. Given a task, we can easily map an action like *pick-up*, *push*, *show*, *rotate*, etc., to a selected box. In fact, we can even order the boxes according to the above criteria. For example, in order to *pick-up* something to place it somewhere else, it may intuitively be a good choice to grasp the *largest* box. When showing the same object to a viewer, it may be better to grasp the *outermost* box.

Similarly, different grasp configurations can be linked to tasks when using a simple representation like a box. We apply another simple mapping from an action to a pre-defined movement here. We already introduced two of these in [1]: the backup power-grasp, which approaches a box until contact, retreats a bit and then closes fingers simultaneously, and the pincher-grasp, which approaches the box until it is in position to closing fingers and contact the box most centrally. One might extend this idea towards the selection of different grasp pre-shapes [16], or even the selection of controllers for different tasks. In fact, Prats *et al.* [17] also use box representations for task-oriented grasping with hand pre-shapes and task frames. However, they assume geometrical knowledge about each object (using a database of 3D models) and structural and mechanical knowledge about a task (e.g. “turning” a door handle).

TABLE I
GRASP COMPONENTS AND OBJECT PROPERTIES

Grasp component	extrinsic properties	intrinsic properties
Transport	position	–
Pre-Grip	orientation, task	size, rough shape
Grip	–	precise shape, weight, surface texture

B. Box Face Visibility

From the box level, we now continue to the face level. Each box provides six rectangular faces in 3D space. Here, we have to consider that incomplete data is produced by a single sensor view of an object, as the back of the object is not visible. Thus, box decompositions are clearly view-dependent and do only envelope visible data points. For this reason, it may be helpful to only take those box faces into account that are visible from the viewpoint. Note that here, “visibility” is understood as the face being oriented towards the viewpoint only, not being visible in sense of occlusion by other objects. We see another motivation for a face visibility check considering the relation between an end-effector, i.e. the robot hand, and the object. Intuitively, humans tend to use grasping movements that involve minimum activity effort. A short experiment at least showed evidence for this:

Test persons had to grasp various objects on a table to describe their appearance, thus the task of grasping was implicit. It showed up that in case of cups, the handle was pinch-grasped when it was orientated towards the human hand, while otherwise the cup body was power-grasped.

Though this experiment is not compelling in terms of a psychophysical evaluation and will therefore not be described any further, it is intuitive in the same way as the viewpoint face check. Valid faces can thereby be selected by being accessible from a given end-effector viewpoint, even if one end-effector might be busy, e.g. holding another object.

In opposition to these observations that a visibility check keeps a large potential, the technical computation if a 3D plane is oriented towards or away from a 3D point is trivial and easy to use. In this way, we also integrate orientation properties into our concept.

C. Box Face Occlusion and Blocking

While the visibility criterion is a check for orientation of faces towards a camera’s or an end-effector’s viewpoint, occlusions and blockings between faces in the box set are also considered. As an example, grasping the head of the duck (Fig. 1) towards the bottom face is not profitable, as this face is “occluded” by a face of the body box. In another way, one may also classify other duck head grasps as being unprofitable. Imagine the 5-finger hand grasping the duck’s head box B_1 from one of the side faces and have in mind that the fingers will not contact the approached face $F_{(1,a)}$, but two of its neighbors, $F_{(1,b)}$ and $F_{(1,c)}$, depending on the grasp orientation. We then define $F_{(1,a)}$ as “blocked” in this grasp orientation, if $F_{(1,b)}$ or $F_{(1,c)}$ is occluded, and remove these grasp hypotheses from the set.

This technique has proven to be very useful in further reducing the number of hypotheses. Technically, the detection of opposing faces is more complex than the visibility check and therefore forms the end of the heuristical selection sequence. Each face of a box has to be compared to each face of all other boxes. The handling of such situations demands an additional computational effort. For this reason, and as it reduces the number of hypotheses drastically, we currently

strictly remove all occluded and blocked hypotheses from our selection.

It may be mentioned that the calculations necessary are purely geometrical problems on faces and points. Like the whole grasp selection process, visibility, occlusion and blocking are currently computed in software (C/C++), one might think about taking advantage of graphical processors to speed up and optimize the geometrical operations.

D. Projection Grids and Learning

The previous steps have been heuristical, aiming at reducing the number of grasp hypotheses according to an object’s task, orientation and shape. Even if it was not named, also the size, i.e. the dimensions of a face, is considered. A face that exceeds the maximum grasp opening in one dimension cannot be grasped. However, there is usually a set of remaining hypotheses from which we would like to select one final grasp. Our current approach to this issue is learning of grasp qualities from 2.5D shape projections.

Considering a box and the points that it envelopes, each face produces a projection of the points onto the face plane. In fact, these projections were already computed for best cut detection [1]. Discretization was made by dividing the face into equally sized cells, thus projections were represented as dynamically sized binary grids. To adapt this representation and enrich it, we now compute linear information, i.e. minimum distance information to the face plane, in a normalized, fixed-sized grid. Fig. 3 shows 18 of such projection grids with size 15×15 for the faces produced by the duck decomposition in Fig. 1. This representation both allows analyzing the 2.5D depth map of each face and fulfills the input space conditions of a classical neural network learner like the one we will use here (see Fig. 2).

In the following experiment three models (homer, mug and duck, see Fig. 1) have been processed by the algorithm and the projections been grasped in the grasp simulator GraspIt! [14]. By providing the two quality measures *eps*, a worst-case epsilon measure for force-closure grasps, and *vol*, an average case volume measure, GraspIt! is automatically used as a teacher for the supervised network, estimating the stability of a grasp on a given face F and its 2.5D projection grid $proj(F)$, respectively. Since due to the normalization in width, height and depth, information about the dimension of F is lost, the box dimensions $dim(F)$ are added in terms of three additional neural network inputs.

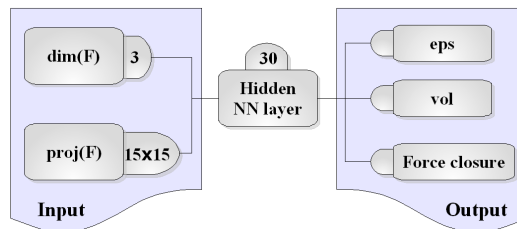


Fig. 2. The neural network structure for off-line learning of grasp qualities from face representations. It holds 228 input, 30 hidden and 3 output neurons. *eps* and *vol* are grasp quality measures that GraspIt! delivers [14]. The force closure is also learned separately even if it equals ($eps > 0$).

IV. EXPERIMENT

We will now present an experiment on the determination of one final grasp hypothesis from a real 3D point cloud. The 3D data is produced from disparity using a stereo vision system, consisting of a Yorick [18] head equipped with two Allied Vision Marlin cameras. The scene is shown in Fig. 4a. As earlier experiments have been performed in simulation only, one focus of the experiment is to test the box decomposition on real 3D data which is influenced by natural dense stereo noise and incompleteness. The second focus is the practical processing of the proposed heuristical and learning selection mechanism, including the considered decisions on task, view-point, shape and size properties.

A. Producing 3D Data

In Fig. 4b, the disparity image produced by the stereo image pair can be seen. It is clearly influenced by incompleteness, both observable by some holes and by the backside which is not visible. Additionally, and though we have cared for a uniform background, there is little noise at the bottom left of the image. The effects of these uncertainties become clearer in Fig. 4c, representing the 3D model of the object.

B. Box Decomposition

We use the box decomposition algorithm [1] to deliver a box approximation of the point cloud. The decomposition steps can be seen in Fig. 6. The fit-and-split algorithm iteratively fits and splits minimum volume bounding boxes, initially starting with the root box enclosing all points (Fig. 6b). The first split, chosen due to maximum volume gain, nicely cuts the outliers from the main shape. The gain parameter Θ^* of 0.41 relates to the new overall box volume being 41% of the box volume before the cut. Out of the two new boxes, the one including the noise keeps to few points and thus is automatically removed. During the following cuts (Fig. 6c-d), the volume gain value increases continuously, since the more the boxes approximate the shape, the less volume can be gained by a cut. After three cuts, the algorithm stops, as a gain threshold below 0.93 will not be reached by any new split. The gain threshold is a parameter of the algorithm and manually set. In practice, threshold values

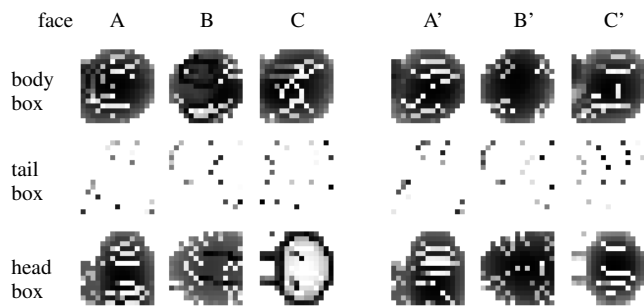


Fig. 3. The set of projection grids for the decomposition in Fig. 1. Three boxes result in 18 faces, where 15×15 grid resolution was chosen. Note that the tail projection is noisy as there are very few points in a very small box. Also note the difference between Head C and C'. C is from below, showing the hollow head, while C' is the projection of the head top.

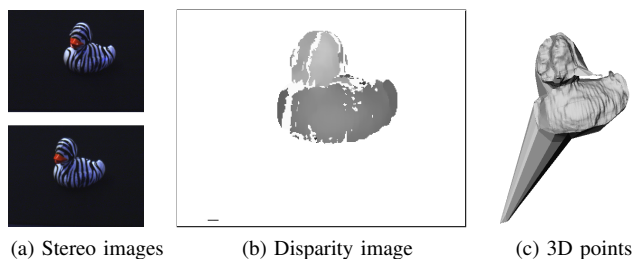


Fig. 4. 3D points from disparity. (a) shows the two images taken from the stereo vision system. From those, disparity values are calculated (b). These can be used to produce 3D points (c). Note that polygonal structures have just been artificially computed with PowerCrust [5] to visualize results. The box decomposition algorithm uses pure 3D point data only.

between 0.9 and 0.95 have led to good results. The higher the threshold, the more cuts will be applied and the more precise the shape will be approximated by boxes.

C. Heuristical Box and Face Selection

With three boxes in the final set, 18 faces and their projections can be accessed. As the decomposition of the real duck is different from the model duck, the box constellations and the projection faces are different. Due to noise and resolution, they are even hard to recognize for the human eye (compare Fig. 5 with Fig. 3 to its left).

As we restrict to grasp orientations parallel to a face's edge, each of the faces theoretically produces four grasps of different orientation. On all 18 faces, this would make 72 theoretical grasp hypotheses available. If we chose a selection of a box by giving an initial task (see Section III-A), as we will do in the following two examples, we could reduce this set to one box with 6 faces, according to 24 grasp hypotheses. The face check selection according to occluded and blocked grasp hypotheses (Section III-C) is presented exemplary for the duck's head box in Fig. 7. As stated, six faces yield four grasp hypotheses each. These rotations are easy to process from one source projection, as the transformation only includes coordinate switching. One face has completely been rejected by occlusion check. It corresponds to the bottom face of the head box. Other faces have been blocked with respect to grasp directions. Intuitively, these are exactly those grasp hypotheses that would cause finger contact on the bottom face, which is

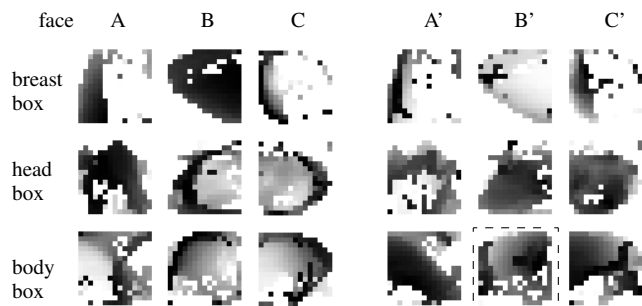


Fig. 5. The set of projection grids for the decomposition in Fig. 6e. Three boxes result in 18 faces, where 15×15 grid resolution was chosen. As the grids are low-resolution, shape is hard to recognize for the eye. One might see the duck pecker facing upwards in head B'. Head B, its opposing face, is visibly a hollow shape. The *pick* grasp is on Body B' (Fig. 8 left).

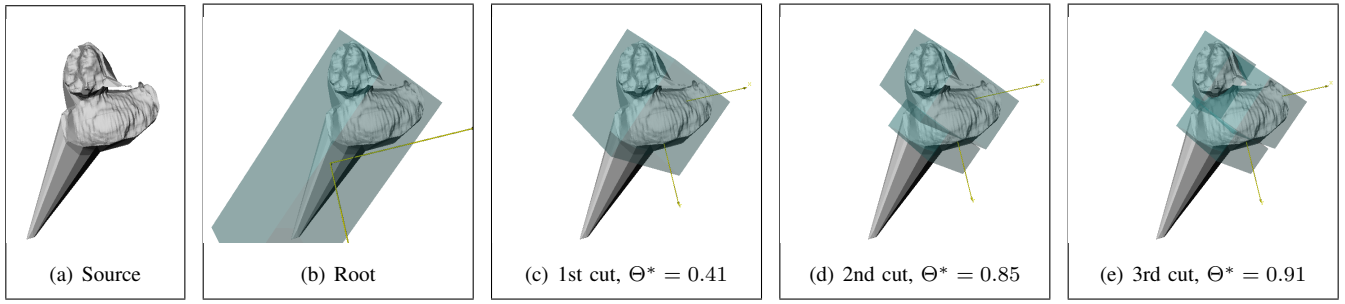


Fig. 6. Decomposition on the source data (a) with a gain threshold of 0.93: (b) The first approximation produces the root box of all points. (c) The first cut separates the noise from the shape. Noise are very few points, so these are not treated further. (d) shows the 2nd cut which still has a good volume gain of 85%. (e) presents the final cut, as further steps did not reach a gain smaller than 0.93.

occluded. For this example, the set of grasp hypotheses is thereby reduced from 24 to 12 hypotheses. Having in mind the option of a viewpoint check as discussed in Section III-B, these could further be reduced to 8, as the head top (C') and only two of the opposing faces (A or A' and B or B') are oriented towards the camera. Note that all the heuristics are optional and not dependent on each other. Using all of them, 72 initial hypotheses were reduced to 8 in this example.

D. Final Grasp Decision and Learning

After having reduced the hypotheses to a small set, we have to finally decide where and how to grasp. The “where” component equals a decision on grasping one of the faces with one orientation. To do this, we apply the neural network structure presented in Section III-D. The face projections of the remaining hypotheses are fed into the net that has been previously off-line trained with artificial examples. After sorting out those hypotheses that do not result in good force-closure response larger than 0.5 (third output), we decide for the one hypothesis with optimal *vol* grasp quality.

Until here, we have not explicitly mentioned the task-dependent decisions (Section III-A). Assume these two tasks and have a look on the corresponding results in Fig. 8:

(T1) *task* : *pick* → *box* : *largest*, *grasp* : *backup*,
 (T2) *task* : *show* → *box* : *outermost*, *grasp* : *pincher*.

The derivation of the final grasp has been performed as presented, where depending on the task, a box selection has been applied. On the final set of hypotheses grids, the one that the trained neural network votes best for is selected. For both examples, these final hypotheses are also marked in Fig. 5 and 7, respectively. Note that in Fig. 5 the selected projection that keeps the best hypotheses is marked for (T1), body box face B' , while in 7 the best hypothesis, head box face C' 90° , is shown for (T2). Additionally, the different choice of grasp type is visible in Fig. 8. In the pick task, the backup grasp focusses on enclosing the whole box, while for the show task the pincher grasp focusses on placing fingers centrally to the contact faces.

In this example, the 3D point cloud had 86310 points, the decomposition algorithm tried 6 fit-and-split iterations, whereof 3 were successful. The decomposition is still the main effort in computation time, it took 22 seconds. The computation of projections as also the heuristical and neural network decisions are neglectable, taking altogether less than half a second. The experiments were performed on a Double Intel Core2 Quad CPU with 2.66 GHz.

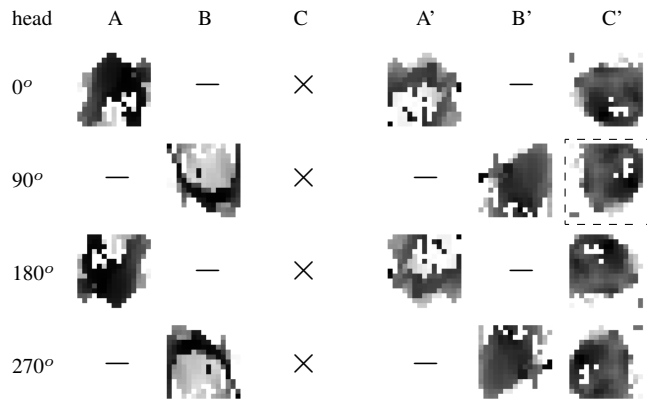


Fig. 7. Face check selection for the duck's head box only. Each head face (columns; see Fig. 5) gives access to four different grasp orientations (rows). Note Head C being completely occluded (\times), as it is the face that connects to the Body box. Some grasp directions are blocked ($-$) from the side. The *show* grasp is on Head C' 90° (Fig. 8 right).

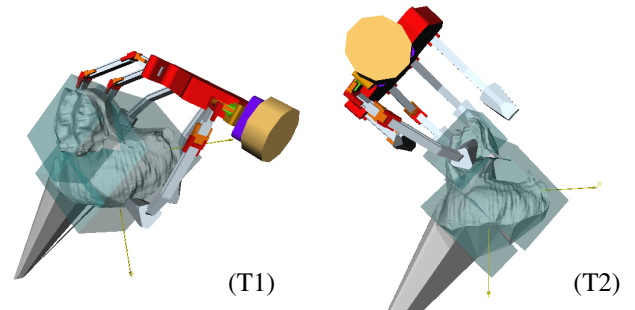


Fig. 8. Left: The final decision for the *pick* grasp (T1). Gripper configuration has been chosen to be the *backup* grasp for enclosing the object. Only faces of the *largest* box (body) have been taken into account. The algorithm finally decided for the top face of the body box, as the neural net forecasted the best grasp quality measure on its projection. Right: The final decision for the *show* grasp (T2). Gripper configuration has been chosen to be the *pincher* grasp for putting fingertips on face center points. Only faces of the *outermost* box (head) have been taken into account. The algorithm finally decided for the top face of the head box, as the neural net forecasted the best grasp quality measure on its projection.

V. CONCLUSIONS

We presented the continuation of box approximation for the purpose of robot grasping. While we specified the core algorithm of box approximation in earlier work, we now concentrated on subsequent steps that all take advantage of the very simple shape representation of boxes. Starting from boxes and their faces that the core algorithm produces, we extended the idea of “grasping on boxes” towards an applicable grasping strategy. This strategy only includes heuristical selection based on efficient geometrical calculations, as also learning from off-line simulation. Basic task-dependencies have been included in this process easily. We see the strength of our approach in its simplicity and its modularity. The simplicity is clear by using boxes and faces in 3D space. Geometric calculations are much more easy to do in contrast to more sophisticated shape primitives like superquadrics. As presented, boxes and faces can additionally take advantage of linear shape projections. The modularity is established by mostly independent criteria and heuristics that complement each other and even leave space for extensions.

There are many possibilities to extend and optimize the current framework both in theory and practice. In theory, we have to evaluate and optimize the current algorithm. Considerations have to be made for the neural net structure, e.g. if it might be better to extend the learning to grasp qualities dependent on the chosen grasp pre-shape, i.e. setting three quality outputs for *each* available grasp pre-shape. Additionally, the simulation part for learning is currently done using static simulation. Thus, contact will stay static between gripper and object, while in dynamics, and reality, the object pose will change dependent on the force applied to it. We are working on this issue also with regard to what we called the grip component. For the sake of efficiency and intuitive motivation, we are aware that our approach is a pre-grip component on very robust shape information. The grip component, as an additional module, would contribute in terms of fine correction based on haptic feedback [13]. In practice, we are still missing some necessary parts to physically perform a grasp with a real robot manipulator. Our current work is also on putting these parts together and connect them to the work proposed here.

The box representation of an object is simple. However, the projection of an object onto the box faces ignores the real 3D shape of the object in the box, not considering the correct surface normals of the object in the grasp planning. Thus, there is a possibility that planned grasps are infeasible, which addresses the limitation of the proposed planning. In future work, we will examine finger positioning estimations on the projections, connected to the work of Morales *et al.* [19]. The effectiveness of the approach in real applications has also to be evaluated through experiments.

As future work, one could also imagine higher-level part classification. Given all three projections of a box, one could try to learn and classify the enclosed shape, which with high probability corresponds to an object part. This relates to work on view-based object (part) representation. Classification of

shape is a beneficial, but also complex task, as additionally, the box constellation might be very different as influenced by noise, perspective view and uncertainties (e.g. compare the different box constellations of the two ducks in Fig. 1 and Fig. 6e). For the purpose of grasping on faces, this is not a very severe problem, while in part and object classification, it probably will be. Therefore, evaluations of these high-level ideas are not a topic of our short-term goal.

VI. ACKNOWLEDGMENTS

This work was supported by EU through the project PACO-PLUS, IST-FP6-IP-027657.

REFERENCES

- [1] K. Huebner, S. Ruthotto, and D. Kragic, “Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping,” in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1628–1633.
- [2] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic Grasping of Novel Objects using Vision,” *Journal of Robotics Research*, 2008.
- [3] E. Lopez-Damian, D. Sidobre, and R. Alami, “Grasp Planning for Non-Convex Objects,” in *International Symposium on Robotics*, 2005.
- [4] E. Lopez-Damian, “Grasp Planning for Object Manipulation by an Autonomous Robot,” Ph.D. dissertation, Laboratoire d’Analyse et d’Architecture des Systèmes du CNRS, 2006.
- [5] N. Amenta, S. Choi, and R. Kolluri, “The Power Crust,” in *6th ACM Symposium on Solid Modeling and Applications*, 2001, pp. 249–260.
- [6] K. Shimoga, “Robot Grasp Synthesis Algorithms: A Survey,” *Journal of Robotic Research*, vol. 15, no. 3, pp. 230–266, 1996.
- [7] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An Overview of Dexterous Manipulation,” in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, 2000, pp. 255–262.
- [8] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, “Automatic Grasp Planning Using Shape Primitives,” in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.
- [9] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, “Grasp Planning Via Decomposition Trees,” in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007.
- [10] G. Barequet and S. Har-Peled, “Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions,” *Journal of Algorithms*, vol. 38, pp. 91–109, 2001.
- [11] J. B. J. Smeets and E. Brenner, “A New View on Grasping,” *Motor Control*, vol. 3, pp. 237–271, 1999.
- [12] N. Derbyshire, R. Ellis, and M. Tucker, “The potentiation of two components of the reach-to-grasp action during object categorisation in visual memory,” *Acta Psychologica*, vol. 122, pp. 74–98, 2006.
- [13] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander, “Demonstration based Learning and Control for Automatic Grasping,” in *Proc. of the International Conference on Advanced Robotics*, 2007.
- [14] A. T. Miller and P. K. Allen, “Graspit! A Versatile Simulator for Robotic Grasping,” *Robotics & Automation Magazine, IEEE*, vol. 11, no. 4, pp. 110–122, 2004.
- [15] T. Baier and J. Zhang, “Reusability-based Semantics for Grasp Evaluation in Context of Service Robotics,” in *Proc. of the International Conference on Robotics and Biomimetics*, 2006, pp. 703–708.
- [16] M. Cutkosky, “On Grasp Choice, Grasp Models and the Design of Hands for Manufacturing Tasks,” *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 269–279, 1989.
- [17] M. Prats, P. J. Sanz, and A. P. Del Pobil, “Task-Oriented Grasping using Hand Preshapes and Task Frames,” in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1794–1799.
- [18] Interactive Systems Research Group (ISRG), University of Reading, “Yorick robot head series.” [Online]. Available: <http://www.isrg.reading.ac.uk/yorick/index.htm>
- [19] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil, “Experimental Prediction of the Performance of Grasp Tasks from Visual Features,” in *Proceedings of the 2003 IEEE/RSJ International Conference on Robots and Systems*, 2003, pp. 3423–3428.

Learning Primitive Actions through Object Exploration

Damir Omrčen, Aleš Ude, Andrej Kos

Jozef Stefan Institute, Slovenia, damir.omrcen@ijs.si, ales.ude@ijs.si, andrej.kos@ijs.si

Abstract— The goal of this paper is to investigate how to acquire useful action knowledge by observing the results of exploratory actions on objects. We focus on poking as a representative type of nonprehensile manipulation. Poking can be defined as a short term pushing action. Here we propose an explorative process that allows the robot to learn the relationship between the point of contact on the object boundary and the angle of poke and the actual response of an object. The robot acquires this knowledge without having any prior knowledge about the action. Initially, the robot was only able to move in random directions. Such self emergent processes are essential for the early cognition. The proposed process has been implemented and tested on the humanoid robot Hoap-3.

I. INTRODUCTION

The main motivation of this work is that many decades of research in the fields of robotics and artificial intelligence (AI) did not result in an intelligent “android like” robot. Why classical AI and robotics did not succeed in building an intelligent robot that can think like a human?

Traditional AI did not succeed due to the lack of a solid theoretical foundation as discussed in a very pointed way by Dennett [1], when he had introduced the “frame-problem”. Additional weakness of the AI in the real world scenarios is the uncertainty in the world information, due to uncertain sensor information. However, in our opinion the most important drawback is that there is no self-emergence in classical AI. The instructor/user has to put more knowledge into the system than he gets it out of it. Nothing emerges by itself. In [2] Lungarella et al. present a survey on developmental robotic, which tries to solve the problems mentioned above.

In this work we investigate how to improve the self-emergence process when learning continuous object-action effects. Our research is part of an EU project PACO-PLUS, whose objective is to develop new methods to endow an artificial robotic system with the ability to give meaning to objects through perception, manipulation, and interaction with people. One of our guiding principles is that new object-action knowledge on a humanoid robot can emerge by exploring the external world. More specifically, by performing actions on different object, the robot can learn the results and preconditions of the actions.

We build cognition on a paradigm of Object-Action Complexes (OAC). Objects and Actions are inseparably intertwined and that categories are therefore determined (and also limited) by the action an agent can perform and by the attributes of the world it can perceive; the resulting, so-called

Object-Action Complexes (OACs) are the entities on which cognition develops (action-centred cognition). Entities (“things”) in the world of a robot (or human) will only become semantically useful “objects” through the action that the agent can/will perform on them. Objects are not just “things” upon which active agents act, but may be able to execute their own actions. Thus each active agent is just another instance of an OAC. This paradigm of OACs offers two novel key issues which will assure that a system with advanced cognitive properties can be developed.

Objects and actions cannot be separated, because objects can induce actions (cup → drink), while actions can redefine objects. While this paper is concerned with OACs at the level of early perception-action events, the project strives to provide a continuous path from such events to complex cognitive processes, where OACs are used as basic building blocks.

To acquire new primitive actions, the robot starts by randomly acting on various objects in its environment. The goal of this explorative process is to acquire new information that was not built into the system. As an example we study how to learn a relatively simple pushing behaviour. We also show how this knowledge can later be used to move (or to control) an object in a desired direction.

Pushing, poking, and rolling are examples of nonprehensile manipulation of objects, i.e. object manipulation without a grasp. This kind of manipulation is used when it is difficult to grasp an object, when an object is too large or too heavy, etc. In this paper we focus on poking as a representative type of nonprehensile manipulation. Poking can be defined as a short term pushing action. Conceptually, our goal is to investigate how to acquire useful action knowledge by observing the results of exploratory actions on objects. For this purpose we study how poking behaviour can be obtained both when the agent generates the exploratory pushes (pokes) and/or when the agent only observes poking actions, performed by other agent or human.

When poking an object, the object motion depends on the object’s shape, weight distribution and on the support friction forces. A lot of work has already been done in the field of mechanics on controllability and planning of poking [5],[6]. Obviously, poking could easily be implemented by assuming a proper representation for the physics of the task, but such an approach relies on a priori knowledge about the action and therefore does not solve the complete learning problem. Additionally, it is sometimes difficult to obtain the model parameters using available sensors (e.g. it is very difficult to obtain friction between the object and the pusher using vision).

If the physical model of the object and the action is not available, the robot has to experiment with different poking actions on the object. In this way the robot acquires new knowledge from exploration and human demonstration in the same way as infants learn their actions – performing actions on objects, i.e. playing with toys. While poking has been used to study cognitive processes before [4], our work focuses on different issues, that is learning complete controllers, whereas Fitzpatrick et al. were primarily concerned with extracting object properties associated with poking actions.

After learning, the robot can use the newly acquired knowledge in order to poke an object in a specified direction. The robot is able to reason how and where an object has to be pushed to move where desired. Our implementation can be divided in two parts. Firstly, the robot learns how an object moves when it is poked from a certain position and from a certain direction. This can be accomplished by experimenting with different poking actions, in which the robot pushes the object several times from different directions and at different locations on the object boundary. During this process the agent builds a knowledge base, which describes the relationship between the point and angle of push on the one side and the actual object movement on the other side. Secondly, the acquired poking knowledge is used to control the object movement, i. e. to push the object along a prespecified trajectory.

II. METHODS

The method for learning poking action has been implemented on a humanoid robot Hoap-3 (Fig. 1). It is 60 cm tall, 9 kg heavy robot equipped with CCD cameras, microphone, foot load sensors and distance sensors. It has 6 DOF in each leg/foot, 1 DOF in the waist, 3 DOF in the neck and 5 DOF in each arm.

As already stated, the goal of the robot is to learn the result of a poking action. It starts by experimenting with different poking actions applied to different objects placed on a table. Afterwards the robot uses the acquired knowledge to reason about the object movement with respect to the performed action. The reasoning should be used later to find the right poking action in order to move the object as desired.

The scene (experiment) has been realized as follows (see Fig. 1). The robot stands at a table and uses a tool to poke the object on the table. The objects used in the experiments are planar polygonal objects. To simplify the environment only one object is placed on a table at a time. To realize one point poking actions, the robot holds a tool in its hand. It is a stick, which increases the robot's workspace. At the end of the tool we have mounted a cylinder, which assures a one point push. The part of the tool which has been used for pushing (the cylinder) will be denoted as a *pusher*. The robot uses only one arm in this experiment. Otherwise, the robot is fixed in the environment. To measure the position and the orientation of the object on the table, the robot uses stereo cameras mounted on its head.

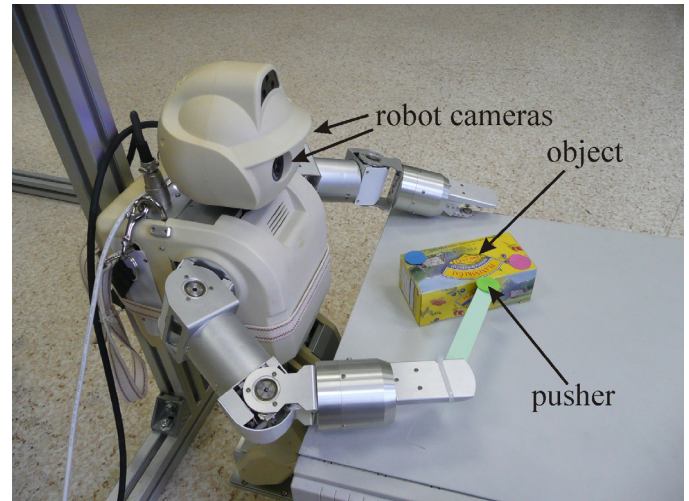


Fig. 1: Robot during pushing action

III. VISION SYSTEM

All objects used for poking were placed on a table. The table is planar, which makes the design of a vision system much simpler. To acquire positions and orientations of the object on the table, it is sufficient to use one camera. We used colour markers to simplify vision processing. We placed two markers on each object in order to extract both position and orientation of the object. Additionally, we marked the pusher to enable visual servoing.

To define the transformation (mapping) between the image coordinate system and the world coordinate system, where the robot is situated in, a calibration has to be performed. The mapping incorporates extrinsic (position and orientation of the camera) and intrinsic (focal lengths, pixel size, image centre) camera parameters. We could estimate the intrinsic and extrinsic camera parameters using other methods (kinematics, chess board...). However, in our case we rather used the robot to move the marker in front of the vision system. Using more than 100 measurements, we calculated the transformation matrix using least-square error methods.

The use of the robot in the calibration process makes the system much simpler and more flexible. Additionally, the result can be more precise, since same data is used during the calibration as well as during poking. So the same sensor uncertainty appears twice and the errors can cancel each other (e.g. kinematics data, vision data...). That means that the same kinematic error which appears during calibration, will also appear during the control – and that will already be included and handled in the calibration process.

The accuracy of the robot and the vision system is rather low. To improve the precision of motion, we had to use visual servoing techniques. To determine the position of the pusher using vision system we put a marker on top of it. Since the vision is calibrated only in one plane, the marker has to lie in that plane. This is true during poking; however, when the pusher is above of the object, the position is not totally correct any more. In this case the robot kinematics is more accurate. To solve this, we have implemented a continuous switch

between kinematics and vision information, where the amount of each depends on the distance from the calibrated plane.

IV. LEARNING

In the first phase of the process, the robot has to learn the behaviour of an object, when the object is poked from a certain direction and at a certain angle (see Fig. 2).

In this phase the robot experiments with different poking actions. The robot has to push an object from different sides of an object as well as under different angles. In the beginning of the process the robot (agent) has no knowledge about the poking action and the robot experiments with different poking actions completely randomly. Afterwards, the robot should only perform action at the points (or angles) where the knowledge (or the model) is not precise enough.

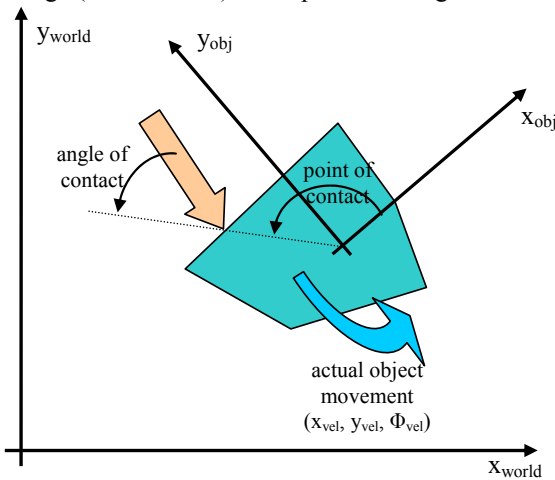


Fig. 2: Schematics of a poking action

After applying a poking action, an object accelerates and changes its position and orientation. Since the objects are very light and the friction between the object and the table is relatively high, we can neglect the dynamic properties of the motion. Typical response of the object is shown in Fig. 3. The object velocity settles in less that 200 ms. The reason for very noisy object velocity is that we have used vision to obtain the position of the markers. However, since training can be done off-line, the data can be filtered and processed before the use.

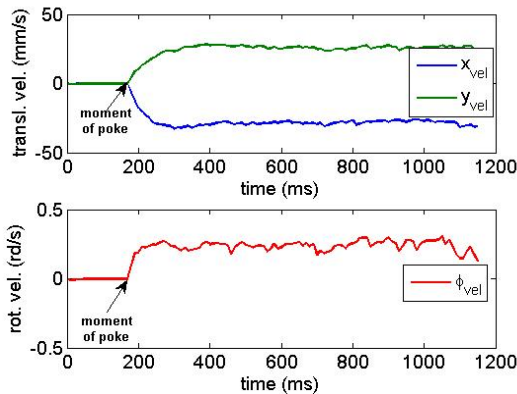


Fig. 3: Typical response (velocity) of an object after applying a poking action

Due to the fact that an object settles its response in a very short time, we can model object behaviour statically. We model the relationship between the displacements of the pusher and the displacement of an object. The displacements of the pusher are expressed by two parameters: the point and the angle of contact on the object boundary. The velocity of the pusher is kept constant. The point of contact is expressed as the angle between the line segment connecting the point of contact and the centre of the object and the x-axis of the object's coordinate system. Similarly, the angle of a contact is expressed as the angle between the pushing direction and the tangent at the point of contact.

The response of an object is represented by three parameters, i.e. the planar velocity of the object centre and the rotational velocity about the centre point on the object. The agent's view of the experiment is shown in Fig. 4.

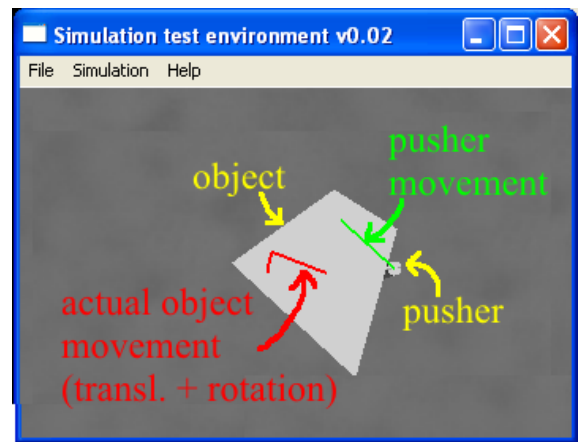


Fig. 4: Agent view of a scene during learning

To represent the relationship between the point and the angle of a contact and the object response, we used a neural network with two hidden layers. Based on the measurements we performed an optimization process and compared neural networks with different number of neurons in each layer. Since we could acquire quite a large set of data, one part of data has been used for learning while the other part of data has been used for verification of the neural network.

The result of the comparison of different neural networks showed that the most reasonable selection is to use different networks for different outputs. The resulting neural networks, which model the object behaviour satisfactory and are still simple enough, are shown in the following table:

	NN inputs	Number of neurons in 1 st hid. layer	Number of neurons in 2 nd hid. layer	NN output
x	position and angle	11	3	Velocity in x dir.
y	position and angle	12	6	Velocity in y dir.
Φ	position and angle	9	4	Velocity in Φ dir.

V. CONTROLLING

After the learning phase is completed, the robot can generate poking actions to move an object in the desired direction. The task of the robot in this phase is to perform a set of poking actions in order to bring an object where desired. Here, a higher-level motion planner should provide the desired movement of the object. The agent has to find out where and how to poke the object to achieve motion close to the desired one. During this process the agent has to use the knowledge acquired in the learning phase. Agent view of the poking scene is shown in Fig. 5.

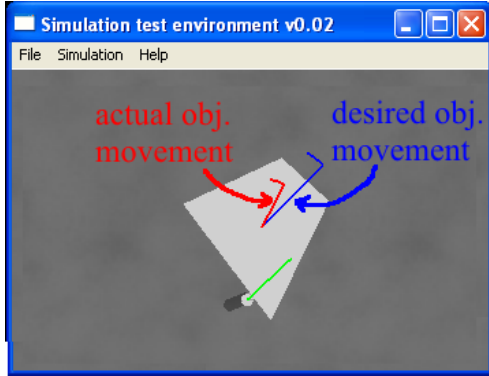


Fig. 5: Agent view of a scene during controlling object movement

Note that, the robot can not always achieve the desired velocity. The desired velocity is or can be defined in three directions (three DOFs); however, the robot controls only two input variables, the point and the angle of contact. Additionally, any arbitrary velocity vector can not be achieved due to the physical limitations of the action (this is still a nonprehensile action).

To achieve optimal motion in a given situation, the agent needs to optimize a criterion function with respect to the point and angle of push, e. g. the weighted square error between the desired motion and the predicted one. Thus we need to find a global minimum of the following function:

$$e = (\mathbf{W}(X_{des} - X_{pred}))^2, \quad (1)$$

where X_{des} represents the desired motion in all three DOFs and X_{pred} represents the motion of the object which is predicted by the neural network, respectively. \mathbf{W} is a weight specifying the importance of each direction.

It is easy to find a local minimum of a function defined in (1) using classical optimisation techniques. However, to find a better solution and to avoid falling into local minima, we run the optimization process several times with different initial values in order to find a better solution or even the global minimum. The solution, which might not be the globally optimal one, results in motion that is usually close to the desired motion. After applying the poking action, the object pose changes and the new point of contact and angle of push are determined, which can be better than the previous ones.

Note that only the directions of object movement are considered in the optimisation. The amplitudes of velocities can be modulated by stronger (faster) pushing action.

VI. RESULTS

The proposed approach has been implemented on a Mitsubishi Pa-10 industrial type robot and on a humanoid robot HOAP-3. The accuracy and the workspace of the Mitsubishi robot are much larger than in the case of the Hoap robot; therefore, it has been much more straightforward to perform and to verify the learning and the control process.

On the other hand the experiments performed on a Hoap robot took us much more time and effort. In the experiments we used only the right arm, which has five DOFs. Technically, to achieve a pushing action with a cylinder (pusher), five DOFs are necessary. Three DOFs are needed to control the position of the pusher and two DOF are needed to control the rotations. One DOF of rotation about the cylinder axis is not important and therefore does not need to be controlled. The robot's right arm also has five DOFs.

Since the robot is rather small and there are no redundant DOFs very small workspace can be achieved. To improve that, we have treated the orientations as less significant and have controlled them in the null space. Additionally, we have used two tools in the same robot hand. One tool has been mounded in such a way that the robot achieved points near the body, while the other tool enables achieving points more far away. At the ends of both tools two cylinders has been mounted, which were used for pushing.

To control the robot we used a velocity based task controller and a quaternion control in the null space for both orientations.

We performed the learning process on a set of different planar objects shown in Fig. 6. Fig. 7 shows the response of a square object in all three directions with respect to the point and angle of contact.

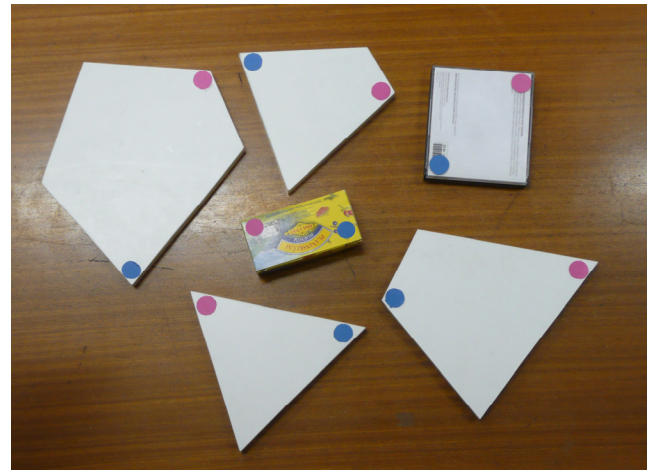


Fig. 6: A set of objects that were used for learning

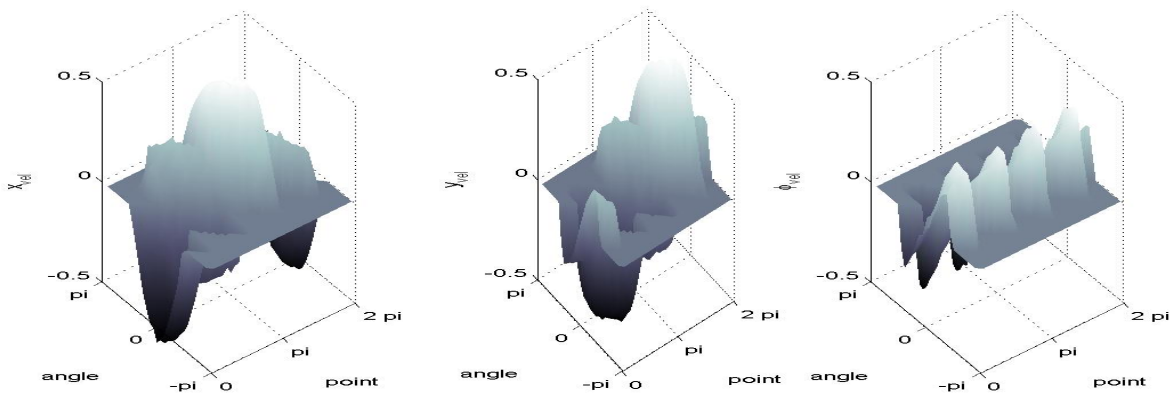


Fig. 7: Relationship between point and angle of contact and object response

In the learning process the robot generated random poking action from all sides of objects (point of contact on the boundary travels from 0 to 2π) and from different angles in the range from $-\pi$ to π . The robot also generated actions that do not result in any object motion (for all angles that are less than $-\pi/2$ or more than $+\pi/2$), where the motion of the pusher is directed away from the object. Based on our experiences we know that such actions do not result in any motion. The learning would be much faster if we provide as much knowledge as possible; however, we wanted that the agent learns this rule by itself, without any hard coding. The goal of our work is to develop a system which could develop cognitive ability of the robot - a system where a robot could evolve in a more intelligent machine. Therefore, such things should not be hardcoded.

To validate the learned controller, we defined a task of consecutive point-to-point movements, where the object orientation was not important. In case of Mitsubishi robot the trajectory has been more complex. The object had to move between the corners of the square of size 30 cm x 30 cm (see Fig. 8). In Fig. 8, points are marked by small circles. Fig. 8 also shows the actual movement of the object (blue line). The object starts from initial position and moves to point P1, then moves through P2, P3 and to P4, and finally returns to P1. The movement of the object is not very precise because the action learning has not been perfect. In any case, we cannot expect that a nonprehensile action would result in a movement with the same precision as an action with a grasp (with full control over an object). Nevertheless, the learned poking action is precise enough to keep the object within a few centimetres of the desired path.

In the case of the Hoap robot the trajectory has been much simpler. The robot had to move a smaller object to a point in space (marked with a red circle in Fig. 9). The trajectory has been defined in such a way that the robot needed to use both tools in order to be able to achieve the task.

Fig. 10 shows the rotation of the object during the whole movement cycle. Since the rotation of the object has not been controlled, it is changing randomly. This was achieved by not including the object rotation in the process of searching the

most appropriate point for pushing. The weight \mathbf{W} was, therefore, set to:

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (2)$$

where 0 as the last diagonal element correspond to the object rotation. Fig. 11 shows the points of contact and the angles of contact during the whole cycle. It can be seen that point and angle of contact change significantly depending on a current object state.

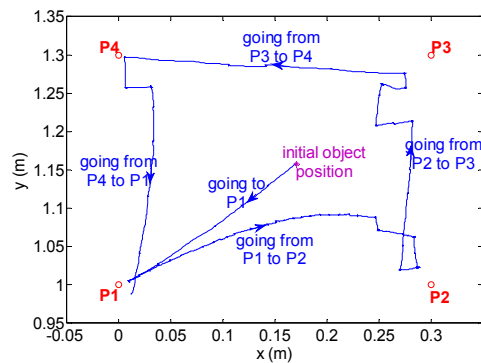


Fig. 8: Object positions during point-to-point movement between the corners of a square (experiments on a Mitsubishi robot)

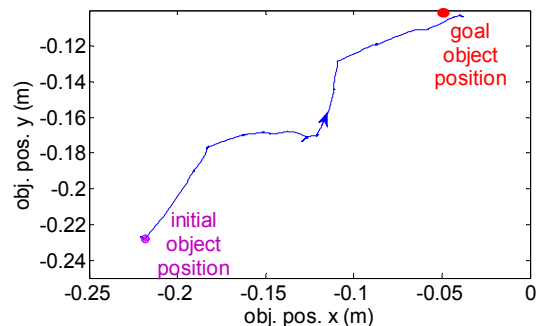


Fig. 9: Object positions during point-to-point movement (experiments on a Hoap-3 robot)

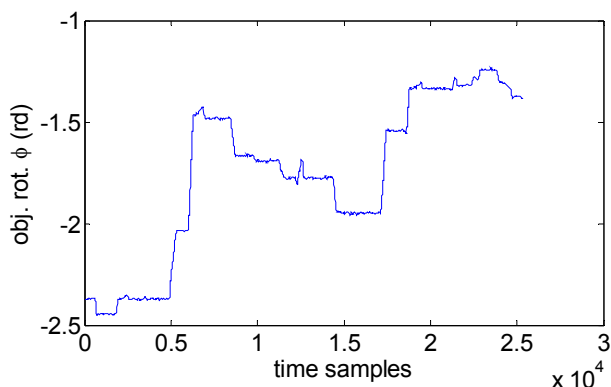


Fig. 10: Object rotation during point-to-point movement

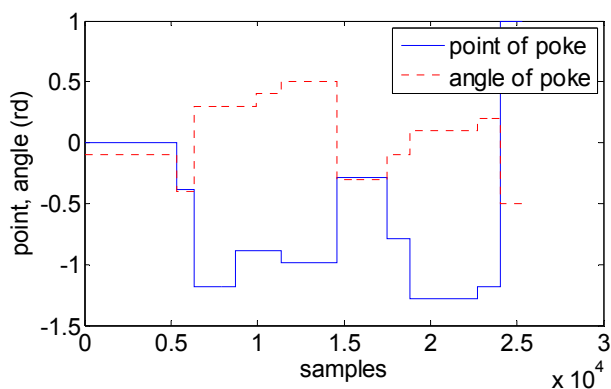


Fig. 11: Point and angle of contact during point to point movement

VII. DISCUSSION AND FUTURE WORK

In this paper we described how to learn the relationship between a point and an angle of a poke and the response of an object. The robot acquired this knowledge by exploration without having any prior knowledge about the action. While very precise learning of pushing actions can take a very long time, the agent learns a rough but reasonable approximation of the action already after a few explorative pushes. This initial knowledge can already be used for a rather rough control of the object movement. Next, while controlling the motion the robot can update its knowledge base by observing the actual movement of the object. Thus the relationship between the desired and the actual object motion gradually becomes more accurate and the control of the object movement direction improves. Additionally, to make the learning of poking actions more optimal, human instructor can demonstrate the most representative pokes (e.g. perpendicular pokes from a few different sides).

However, the knowledge, that the robot obtained by exploration, is useful only for the object that was used for training. Currently, for each new object exploration has to start from the beginning, thus it takes a long time before a satisfactory large object library is built. There is no

generalisation. Our plan for the future is to learn more general pushing controllers instead of learning the behaviour of every object. The generalisation can be achieved by performing many different pushing actions on different objects. The actions and object has to differ in relevant characteristics in order to identify the general pushing rule. To solve such problems, some authors use the recurrent neural networks with parametric bias [7],[8]. In these works, static images of objects are linked to dynamic features of objects.

Using such general laws, people can predict the movement. However, when the actual movement of the object differs from the predicted one, humans include the feedback loop and adapt their actions in order to achieve the desired motion of the object. In the same way closed loop control has been used in our work. The robot/agent can predict only the approximate behaviour of the object. Due to the object properties that has not been modelled or cannot be measured, e.g. friction, mass distribution, etc., the actual motion differs and the robot has to adapt its motion to improve the motion of an object.

In summary, we realized the process of associating object-action events through an explorative, self emergent process. Such processes are of great importance for the early cognition. No knowledge about pushing was provided to the robot. We only provided rules about how to explore the environment and the robot obtained the controller by itself.

ACKNOWLEDGMENT

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657).

REFERENCES

- [1] Dennett, D. C., "Cognitive wheels: The frame problem of AI," In Hook-way, C., editor, *Minds, machines and evolution*, 129–151. Cambridge University Press, 1984.
- [2] M. Lungarella, G. Metta, R. Pfeifer and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151 – 190, 2003
- [3] Integrated project proposal "Perception, Action, and Cognition Through Learning of Object-Action Complexes (PACO-PLUS), Cognitive Systems, FP6-2004-IST-4-2.4.8.
- [4] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning About Objects Through Action - Initial Steps Towards Artificial Cognition", *Proc. 2003 IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, pp. 3140-3145, September 2003.
- [5] K. M. Lynch and M. T. Mason, "Stable pushing: mechanics, controllability, and planning," *The international journal of robotics research*, vol. 15, no. 6., pp. 533-556, 1996
- [6] Q. Li and S. Payandeh, "Manipulation of convex objects via two-agent point-contact push," *The international journal of robotics research*, vol. 26, no. 4, pp. 377-403, 2007
- [7] S. Nishide, T. Ogata, J. Tani, K. Komatani and H. G. Okuno, "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," *IEEE Int. conf. on Rob. and Autom.*, pp. 2501-2506, Italy, 2007
- [8] J. Tani and M. Ito, "Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment," *IEEE Trans. on SMC Part A*, Vol. 33, No. 4, pp. 481-488. 2003

A Strategy for Grasping unknown Objects based on Co-Planarity and Colour Information

Mila Popović¹, Dirk Kraft¹, Leon Bodenhagen¹, Emre Başeski¹,
Nicolas Pugeault¹, Danica Kragic², and Norbert Krüger¹

¹*Cognitive Vision Lab,
The Mærsk Mc-Kinney Møller Institute,
University of Southern Denmark,
Campusvej 55, DK-5230 Odense, Denmark
Email:{mila, kraft, emre, nico, norbert}@mmmi.sdu.dk, lebod04@student.sdu.dk*

²*Centre for Autonomous Systems
Computational Vision and Active Perception Lab
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
dani@kth.se*

Abstract

In this work, we describe and evaluate a grasping mechanism that does not make use of any specific object prior knowledge. The mechanism makes use of second-order relations between visually extracted multi-modal 3D features provided by an early cognitive vision system. More specifically, the algorithm is based on two relations covering geometric information in terms of a co-planarity constraint as well as appearance based information in terms of co-occurrence of colour properties. We show that our algorithm, although making use of such rather simple constraints, is able to grasp objects with a reasonable success rate in rather complex environments (i.e., cluttered scenes with multiple objects).

Moreover, we have embedded the algorithm within a cognitive system that allows for autonomous exploration and learning in different contexts. First, the system is able to perform long action sequences which, although the grasping attempts not being always successful, can recover from mistakes and more importantly, is able to evaluate the success of the grasps autonomously by haptic feedback (i.e., by a force torque sensor at the wrist and information about the distance of the gripper after a grasping attempt). Such labelled data is then used for improving the initially hard-wired algorithm by learning. Moreover, the grasping behaviour has been used in a cognitive system to trigger higher level processes such as object learning and learning of object specific grasping.

Key words: Vision based grasping, Cognitive systems, Early cognitive vision

1. Introduction

The capability of robots to effectively grasp and manipulate objects is necessary for interacting with the environment and thereby fulfil complex tasks. These capabilities need to be implemented and evaluated in natural environments, considering both known and unknown objects. Considering the important requirements for the next generation of service robots such as robustness and flexibility, robots should be able to work in unknown and unstructured environments, be able to deal with uncertainties in feature acquisition processes as well as to work fast and reliable. These requirements also assume that the robots are able to

deal with initially unknown objects as well as to be able to learn from experience. The work introduced here describes an algorithm for grasping of unknown objects as well as the improvement of this algorithm through learning. The basic idea is the modelling and generation of *elementary grasping actions* – simple perception-action pairs suitable for generation of grasps where very little or no information about the objects to be grasped is known *a-priori*.

The body of work in the area of robotic grasping is significant (see, e.g., [1–10]). We distinguish approaches based on the level of *a-priori* object information used to model the grasping process. In particular, objects to be grasped may be assumed to be known, that is, both the shape and

the appearance of the object are known and used to associate specific grasping strategies to them through exploration, (see, e.g., [2,3]) or different types of supervised learning (see, e.g., [9,10]). When objects are assumed to be unknown, the assumptions of the system naturally need to be much more general in order to generate suitable grasping hypotheses (see, e.g., [4]).

In this work, we describe an adaptable grasping algorithm in which no prior object knowledge is used in the beginning but which is used to establish grasping making use of such object knowledge. More specifically, we

- 1) define an initial grasping behaviour not requiring any prior object knowledge. This behaviour is based on coplanar contours extracted by the early cognitive vision system [11,12] and already shows a high success rate in complex scenes,
- 2) refine this initial grasping behaviour by supervised learning in which however the labelling of the training data is done automatically during an exploration process, and
- 3) use the initial object independent grasping behaviour to constitute object shape representations (see also [13]) and associate grasping affordances to those¹ and hence lead to grasping based on object prior knowledge.

The paper is organised as follows. We first give an overview of the state of the art in robot grasping in Section 2, where we also outline distinguishing features of our approach in comparison to existing work. In section 3 we describe the visual representations from which the grasps become computed. In section 4, we describe how the grasps are computed from the visual features. The experimental setup and the evaluation of the grasping strategy is described in section 5. In section 6, the role of the grasping strategy in the cognitive system is discussed. In particular, we discuss the aspects of fine-tuning the grasping strategy by learning and the application of the grasping strategy in a bootstrapping process wherein objects and grasp knowledge thereof become acquired by autonomous exploration.

2. Related work

In this section, we present an overview of the current research in the area of robotic grasping and relate it to our work.

One area of research in the field of object grasping are analytical approaches (see, e.g., [15,5–7]) that model the interaction between a gripper and an object to compute promising grasps. When contact points between the robot hand and the object are determined and the coefficients of friction between the two materials are known, it is possible to calculate a wrench space - i.e., 6D space of forces and

¹ Here we only briefly describe the role of the initially object independent grasping behaviour in object knowledge based grasping. Its use for such grasping requires additional complexities such as object memory, pose estimation and a probabilistic representation of object-grasp associations that are beyond the scope of this paper. These are fully treated in a separate publication (see [14]).

torques that can be applied by the grasp. A force-closure grasp can resist all object motions provided that the gripper can apply sufficiently large forces. These forces can be measured by tactile sensing (see e.g. [8]) and grasp quality can be computed as objective functions which can be further enhanced by optimising the parameters of a dextrous hand (see, e.g., [16,17]). In most of those approaches it is assumed that either the shape properties of the object are known or that these can be easily extracted using visual information which can be difficult in realistic settings.

Related to the analytical approaches are considerations on the robot embodiment. Since robot hands often have many degrees of freedom, the search space of possible grasp configurations is very large. Analytical approaches are therefore usually used together with some heuristics which guide and constrain the optimisation process. For example, heuristically-based grasp generators often include some grasp preshape types (see, e.g., [18,19,4]) based on human grasping behaviour. Domain specific knowledge, e.g. workspace constraints, hand geometry, task requirements or perceptual attributes are also used (see, e.g., [20,21,17]). In addition, simulations can further speed up the learning process (see, e.g., [22,23]).

In industrial applications, the association of grasps to known objects is often done manually or by guiding the gripper directly to an appropriate pose during a training phase where the object is in a known pose. Learning by demonstration (see, e.g., [24–26]) can be a very efficient tool to associate grasps to known objects, in particular when dealing with humanoid robots. Once prior knowledge is present in terms of a 3D object model and defined grasping hypotheses (see, e.g., [27]), the grasping problem is basically reduced to object recognition and pose estimation.

Another approach is learning by exploration. In the recently submitted work [14], grasp densities become associated to 3D object models which allow for memorising object-grasp associations with their success likelihoods. In this context, a number of learning issues become relevant such as active learning (see, e.g., [28]) and the efficient approximation of grasp quality surfaces from examples (see, e.g., [9]). An interesting approach, which can be positioned inbetween grasping with and without object prior knowledge, is the decomposition of a scene into shape primitives to which grasps become associated (see, e.g., [17,18]).

Grasping unknown objects is acknowledged to be a difficult problem which varies in respect to the complexity of objects and scenes. Many projects (see, e.g., [29,4,10]) share the following sequence of steps S1–S4:

- S1 Extracting relevant features
- S2 Grasp hypotheses generation
- S3 Ranking of grasp hypotheses
- S4 Execution of the best candidate grasp

The complexity of a system depends on the choice of sensors, the diversity of considered objects, the scene configuration and the kind of a-priori knowledge assumed. A number of examples relies on visual sensors and a simple gripper with 2 or 3 fingers [30–32,4]. In [32–34] the 2D contours of

an object are used as a relevant feature and grasp planning as well as quality evaluation are based on approximating the centre of mass of the object with the geometrical centre of the contour. Often the camera is positioned above the scene, pointing vertically down and in some cases several object contours were captured from different angles [30]. Most contemporary vision based approaches assume a simple situation where the scene consists of one object placed against a white background, such that the segmentation problem is minimal. Other approaches use range scanning sensors, [35–37]. This is an attractive choice, since they provide detailed geometrical model of an object. When a detailed 3D model is available the grasp planning does not differ a lot from the case of grasping known objects.

Some recent work considers also the generation of grasping hypotheses based on local features rather than the object shape model [10]. The algorithm is trained via supervised learning, using synthetic images as training set. From two or more images in which grasping hypotheses are generated, the system performs approximate triangulation to derive 3D position of the grasping point. The work of [38] makes use of explicit information in terms of 2D position and orientation to learn feature combinations indicative for grasping. The tasks of computing such feature combinations can be linked to the concept of ‘affordances’ proposed by Gibson [39]: The occurrence of a certain feature combination potentially triggers a certain grasping action indicating the ‘graspability’ of the object. A challenging task is to learn such object affordances in a cognitive system (see, e.g., [40]). Our work does not rely on object specific prior learning but it can generate the grasp hypotheses based on the current relationship between scene features. In particular, our system uses 3D features which can provide more optimal grasps in terms of approaching the object and orienting the hand accordingly.

Once a contact with the object is made, tactile information can be used to further optimise the grasp (see, e.g., [8,41,42]). In [41], a data-base that matches a tactile information patterns to successful grasps is used to guide the grasping process. Self-Organizing Maps are used for the interpolation of grasp manifolds associated to shape primitives. In [8], so called ‘Contact Relative Motions’ (CRMs) triggered by tactile information are used to translate the grasp synthesis problem into a control-problem with the aim of finding the shortest sequences of CRMs to achieve stable grasps. Our prior work presented in [42] shows how tactile feedback can be used for implementation of corrective movements and closed loop grasp adaptations. In this work, we show how tactile feedback can be used to confirm the success of an executed grasp.

Note that some initial work on our approach described here has previously been presented at a conference [43] where the system was tested only in simulation and thus did not deal with any real-world problems. In the work presented here, we have implemented the grasping system on an actual hardware consisting of a stereo vision system and a robot arm. As a consequence of the extensive evaluation

done here, it was required to make a number of significant modifications compared to [43]. Moreover, we have introduced an adaptive component in our approach and discuss the work in the context of a concrete cognitive system.

2.1. Contributions and relation to prior work

As outlined in the previous section, grasping of unknown objects in unconstrained environments is a hard problem due to the small amount of prior knowledge that can be assumed. To create a system that solves this problem in a general way with high success rate, a number of strategies need to become combined and learning needs to be an integral part of such a system. Our algorithm provides a strategy based on 3D edges and other visual modalities and can be seen as being complementary to strategies based on 2D features or 3D descriptions extracted by range scanners. Here, we point out specific contributions of our work related to the existing grasping approaches.

- D1 **Weak prior knowledge:** Our grasping strategy is based on a weak prior information of objects to be grasped: In particular, it is based on the existence of co-planar pairs of 3D edges. We will show that such basic cues can already lead to a large amount of successful grasps in complex scenes (see D4) and hence can be used in a bootstrapping process of a cognitive system in which stronger bias is developed by experience (see D6 and [13]).
- D2 **3D representation:** Our approach makes use of the full potential of 3D information. The prior knowledge we use generates a full 3D pose and hence we can also grasp objects that are tilted in any 3D orientation (see figure 12).
- D3 **Error recovery:** Because of the weak prior we can not expect our approach to work with a success rate close to 100%. We prefer to generate a certain percentage of successes on arbitrary objects rather than high quality grasps on a constrained set of objects. However, for this the system needs to be able to continue in case of unexpected events and non successful grasps (see figures 8, 9 and 12).
- D4 **Applicable on difficult scenes:** Most work in grasping is based on ‘single grasp attempt/single object’ situation. In contrast, we will work on rather complex scenes with multiple objects and no pre-segmentation. We can show that even in such scenes, we have a reasonable success rate. Moreover, due to the error recovery (D3) we are able to perform full sequences of grasping attempts (see [44] for a movie).
- D5 **Autonomous success evaluation:** We can confirm the success by means of haptic information. By that, we are able to building up an episodic memory (see figure 11) of evaluated grasping attempts, containing: 1) the grasping hypotheses, 2) the visual features that generated them, and 3) a success evaluation. These triplets are used as a ground truth for further learning and fine-tuning (see D6).

D6 **Memorisation and learning:** The autonomously generated ground truth is stored in an episodic memory and is used as input for a learning based on neural networks to refine the pre-wired grasping strategy.

D7 **Realisation on different embodiments:** We show that the grasping behaviour can be realised on different embodiments. More specifically, we applied it with a two-finger gripper as well as a three finger hand.

3. Visual representation

The grasping behaviour described in this work is based on the early cognitive vision system [11,12]. We use a calibrated stereo camera system to create sparse 2D and 3D features, namely multi-modal primitives (described in Section 3.1), along image contours. In this system, we compute local information covering different visual modalities such as 2D/3D orientation, phase, colour, and local motion information. This local information is then used to create semi-global spatial entities that are called contours (described in Section 3.2). In Section 3.3 two perceptual relations, co-planarity and co-colourity are defined between primitives and between contours, and later used in calculation of grasping hypotheses. Note that primitives, contours and their perceptual relations are particularly important in the context of this work, since the grasping hypotheses defined in Section 4 are based on them.

3.1. Multi-modal primitives

2D primitives represent a small image patch in terms of position \mathbf{x} , orientation θ , phase ϕ and three colour values ($\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r$) describing the colour on the left and right side of the edge as well as on a middle strip in case a line structure is present. They are denoted as $\pi = (\mathbf{x}, \theta, \phi, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r))$. Pairs of corresponding 2D features across two stereo views afford the reconstruction of a 3D primitive encoded by the vector

$$\Pi = (\mathbf{X}, \Theta, \Phi, (\mathbf{C}_l, \mathbf{C}_m, \mathbf{C}_r))$$

in terms of a 3D position \mathbf{X} and a 3D orientation Θ as well as phase and colour information generalized across the corresponding 2D primitives in the left and right image (for details, see [12]).

Figure 1 illustrates what kind of information exists on different levels of the feature extraction. The process starts with a pair of stereo images (figure 1 (a)). Then the filter responses (figure 1 (b)) are calculated which give rise to the multi-modal 2D primitives and contours (figure 1 (c)). After finding corresponding 2D feature pairs across two stereo views, the 2D information is used to create 3D primitives and 3D contours (figure 1 (d)).

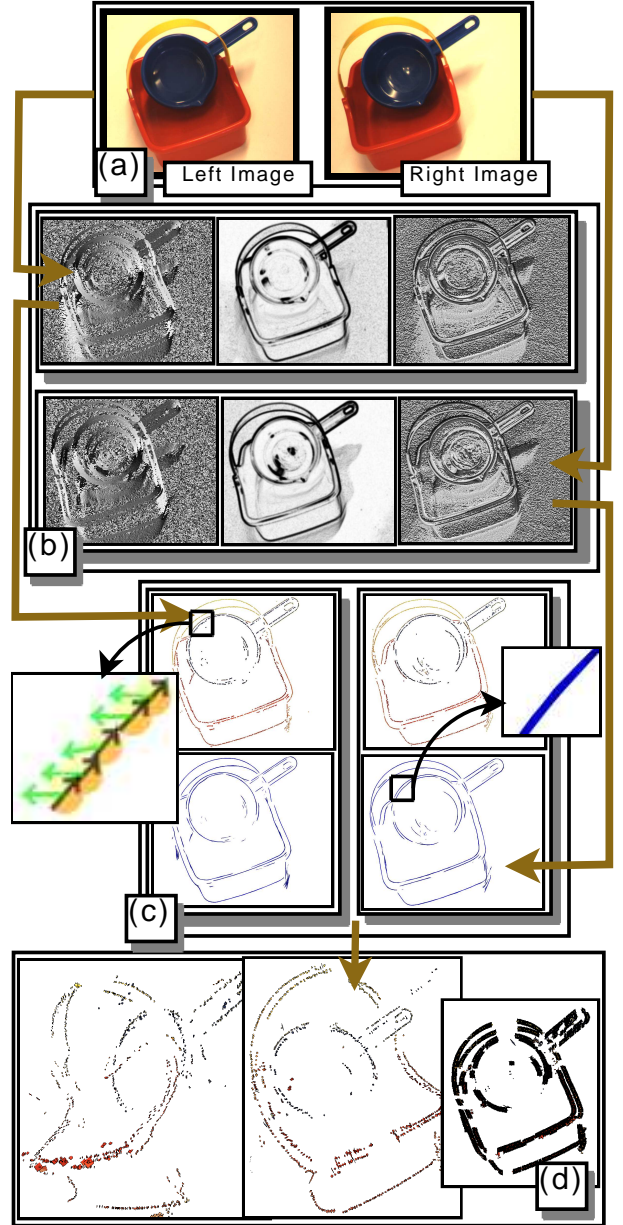


Fig. 1. Different type of information that is available in the representation. (a) Original stereo images. (b) Filter responses. (c) 2D primitives and contours. (d) 3D primitives from two different view points and 3D contours.

3.2. Contours

Collinear and similar primitives are linked together by using the perceptual organisation scheme described in [45] to form structures denoted as *contours*. Since the linking is done according to geometrical and visual good continuation, contours represent parts of a scene as geometrically and visually smooth curves. As their building blocks, contours are also multi-modal entities containing visual modalities such as mean colour and phase. Therefore, they do not only contain geometrical but also appearance based information. In figure 2, 3D contours of an example scene are presented.

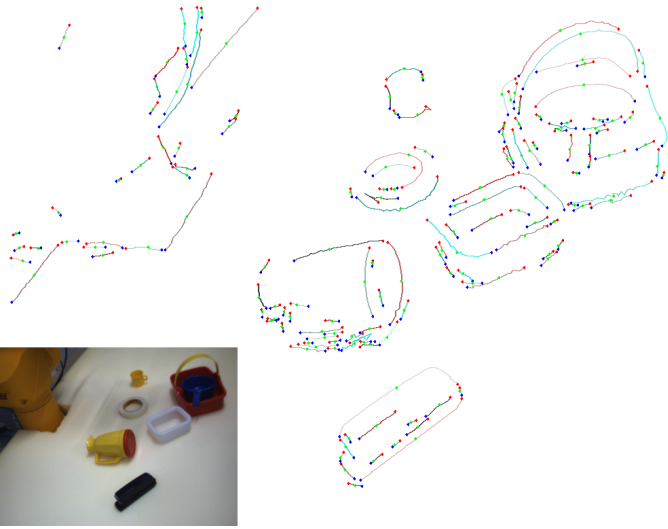


Fig. 2. 3D contours extracted from the scene that is shown in the bottom left (left image). Red dots indicate the first primitive in a contour, green the middle, and blue the last primitive in the contour.

3.3. Relations between primitives and contours

The sparse and symbolic nature of the multi-modal features gives rise to perceptual relations defined on them that express spatial relations in 2D and 3D (e.g., co-planarity, co-colourity). The co-planarity relation (see figure 3b) between two spatial 3D primitives Π_i and Π_j is defined as:

$$cop(\Pi_i, \Pi_j) = \frac{\Theta_j \times \mathbf{V}_{ij}}{|\Theta_j \times \mathbf{V}_{ij}|} \bullet \frac{\Theta_i \times \mathbf{V}_{ij}}{|\Theta_i \times \mathbf{V}_{ij}|} \quad (1)$$

where \mathbf{V}_{ij} is the vector connecting the two primitives positions.

Two 3D primitives are defined to be co-colour if their parts that face each other have the similar colour. Note that the co-colourity of two 3D primitives is computed using their 2D projections. We define the co-colourity (see figure 3 (a)) of two 2D primitives π_i and π_j as:

$$coc(\pi_i, \pi_j) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j), \quad (2)$$

where \mathbf{c}_i and \mathbf{c}_j are the RGB representation of the colours of the parts of the primitives π_i and π_j that face each other; and $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is the Euclidean distance between RGB values of the colours \mathbf{c}_i and \mathbf{c}_j .

Since contours represent larger portions of scenes than local features, contours and their relations can give a more global overview of the scene. The contour relations used in this work are straightforward extensions of primitive relations. While calculating relations between two contours, the primitives that create the contours are associated between the contours and the relations are calculated as the mean relations between associated primitives.

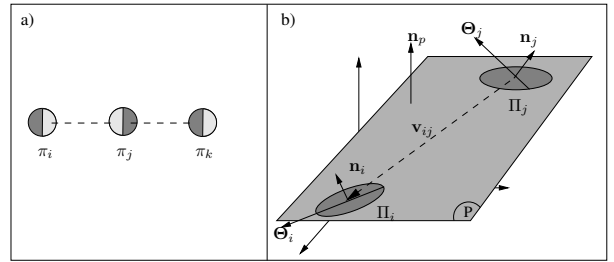


Fig. 3. Illustration of the perceptual relations between primitives. **a)** Co-colourity of three 2D primitives π_i, π_j and π_k . In this example, π_i and π_j are co-colour, so are π_i and π_k ; however, π_j and π_k are not co-colour. **b)** Co-planarity of two 3D primitives Π_i and Π_j . \mathbf{n}_i and \mathbf{n}_j are normals of the planes that are defined as cross products of individual primitives orientations Θ_i, Θ_j and the orientation of the connecting line \mathbf{V}_{ij} , (see sec. 4.1). \mathbf{n}_p is the normal of a common plane defined by combining the two normals \mathbf{n}_i and \mathbf{n}_j .

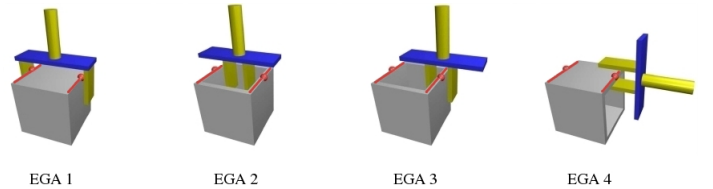


Fig. 4. Elementary grasping actions (EGAs), figure adapted from [43]. The red lines indicate 3D edges that have been reconstructed from stereo images. They appear in pairs, and represent the pair of contours that are connected by relations of co-planarity and co-colourity. The red dots represent the 3D primitives in the middle of each contour. In case of EGA 2, the gripper fingers are initially closed and the grasp is accomplished by opening fingers and thus applying force to the concave objects from inside out. EGA types 3 and 4 each generate two actions, one for each parent primitive. See also figures 10 and 12.

4. Grasping Strategy

The grasping behaviour proposed in this work is a low-level procedure that allows for the robot manipulator to grasp unknown objects. As explained in section 3, the early cognitive vision system extracts multi-modal visual feature descriptors from stereo images. Multi-modal relations between primitives support perceptual grouping. Second order relations, co-planarity and co-colourity, between contours indicate possible co-planar edges originating from the same object, or even the same surface in a scene. The grasping behaviour is based on four basic grasping actions that can be performed on a pair of such contours using a parallel gripper. In the early cognitive vision system, edges are represented as 3D contours. As described in Section 3, 3D contours are sets of the linked 3D primitives. The pair of contours that are both co-planar and co-colour are called "similar contours". For each of the two similar 3D contours, one representative 3D primitive is chosen. These two primitives are called 'parent primitives' and they contain the information about respective contour's position and orientation. Figure 4 shows the four types of elementary grasping actions (EGAs) defined by two parent 3D primitives.

It is important to notice that in a real scene only some

of the four suggested grasps are meaningful. For example, if an object in the scene is not concave, only grasps of type EGA1 can be successfully performed. Since the information provided by the initial image representation is not sufficient to determine which of the grasping actions are suitable, the system suggests grasps of all four EGA types. Suggested grasping actions are therefore called *grasping hypotheses*. The term is also appropriate since grasping actions can fail because of other factors (such as uncertainties in the position and the orientation of the gripper that come from the uncertainty of the visual reconstruction, from limitations of the manipulator, or from an unforeseen collision with the environment) even if the intended action was reasonable.

4.1. Elementary Grasping Actions (EGAs)

Two parent primitives Π_i, Π_j produce a set of parameters used for defining the four EGAs. The parameters (see figure 3) are given as follows:

- position and orientation of the common plane p defined by co-planar parent primitives. It is denoted by position \mathbf{P}_p of the point in the common plane half way between \mathbf{X}_i and \mathbf{X}_j and orientation \mathbf{n}_p of the plane normal
- distance between parent primitives: $d_p = \|\mathbf{V}_{ij}\|$ (figure 3)
- direction connecting the parent primitives: $\mathbf{D} = \frac{\mathbf{V}_{ij}}{d_p}$
- individual primitives orientations Θ_i and Θ_j

This section starts with the definition of the common plane p and then proceeds to show how specific EGA types are constructed.

The common plane p is represented by \mathbf{P}_p and \mathbf{n}_p which are calculated as:

$$\mathbf{n}_p = \pm \frac{\Theta_i \times \mathbf{D} + \Theta_j \times \mathbf{D}}{\|\Theta_i \times \mathbf{D} + \Theta_j \times \mathbf{D}\|} \quad (3)$$

$$\mathbf{P}_p = \frac{\mathbf{X}_i + \mathbf{X}_j}{2}$$

where \mathbf{X}_i is the position of the i_{th} 3D primitive in the scene. Note that we assure that $(\Theta_i \times \mathbf{D}) \cdot (\Theta_j \times \mathbf{D}) > 0$ by choosing the direction of the Θ_j , so that vectors $\Theta_i \times \mathbf{D}$ and $\Theta_j \times \mathbf{D}$ point into similar directions.

The plus-minus sign on the right hand side of the equation above indicates that the direction of the normal of the averaged plane is also arbitrary. It is important to know which direction of the plane normal to use in order to predict meaningful grasps. The initial scene representation does not provide this information. Nevertheless, it is intuitively clear to the human viewer why the top side of the box on figure 4 (EGA 1) should be grasped from above. This observation can be expressed mathematically. The normal of the visible side of a surface always forms an obtuse angle to the vector originating from the point of view and pointing to the surface (figure 5). When this observation is turned around, it follows that visible surfaces should adopt the direction of the normal that forms an obtuse angle to the camera ray in order to give expectable grasps. Another

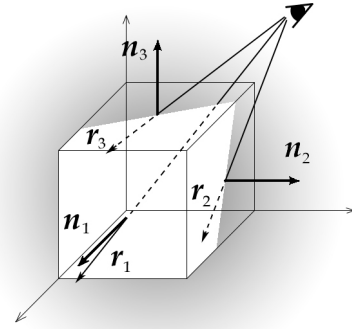


Fig. 5. Choosing the correct surface normal. $\mathbf{n}_1, \mathbf{n}_2$, and \mathbf{n}_3 are outward surface normals marking the sides of the cube visible on the illustration. The two sides visible from the marked point of view have surface normals \mathbf{n}_2 , and \mathbf{n}_3 . $\mathbf{r}_1, \mathbf{r}_2$ and \mathbf{r}_3 are camera rays, vectors originating from the marked point of view and pointing to the surface normals.

aspect of this observation concerns camera placement. Visible features of an objects should be the ones reachable by the manipulator. This kind of reasoning is applicable for EGA 1, EGA 2 and EGA 3 cases, while EGA 4 type of grasp does not depend on the direction of the plane normal but still requires that only one direction is adopted as the opposite direction would only duplicate already existing hypotheses.

Using the argumentation above, we adopt a heuristics where only one direction of normal is used for generating EGAs. The advantages are that the number of produced hypotheses is dramatically reduced (number of EGA 1, 2 and 3 grasps is halved), and in the majority of cases the wrong hypotheses are excluded.

4.1.0.1. *Mathematical formulation of EGAs* A grasp is defined by the position and the orientation of its tool reference frame (Tool Centre Point (TCP) reference frame) in relation to, for example, the Robot's Base reference frame (figure 6), and the initial distance d between gripper fingers.

If the origin and the orientation of the TCP reference frame are defined as in figure 6 such that \mathbf{Z}_{TCP} (Z axis of TCP frame) is parallel to the gripper's fingers, \mathbf{X}_{TCP} axis connects the fingers, and $\mathbf{Y}_{TCP} = \mathbf{Z}_{TCP} \times \mathbf{X}_{TCP}$, and the origin is placed between two fingers, on some negative \mathbf{Z}_{TCP} distance (depth of the grasp) from fingertips, then elementary grasping actions are given with expressions as follows.

EGA 1:

$$\begin{aligned} \mathbf{P}_{TCP} &= \mathbf{P}_p \\ \mathbf{Z}_{TCP} &= -\mathbf{n}_p \\ \mathbf{X}_{TCP} &= \mathbf{D} \\ d_p &< d \leq d_{max} \end{aligned} \quad (4)$$

Initial finger distance d should be bigger than the distance between parent primitives d_p , so that grasping position can

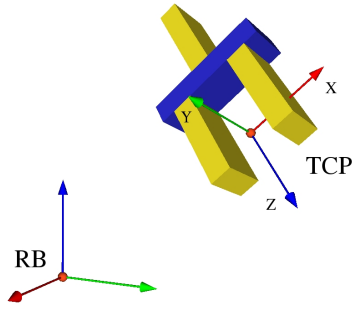


Fig. 6. The figure shows the Tool Centre Point (TCP) reference frame, it is given in respect to the robot's base (RB) frame. The position and orientation of the TCP reference frame is used when defining elementary grasping actions.

be approached without colliding with the object. It is limited by the maximum fingers opening distance d_{max} . The \mathbf{X}_{TCP} can have the opposite direction as well ($-\mathbf{D}$) when using a parallel gripper, as the gripper has reflection symmetry across ZY plane.

EGA 2: is a grasp that is designed for concave objects, it has same the position and the orientation as *EGA 1* but the initial finger distance is zero and fingers are opened in order to grasp an object (figure 4, *EGA 2*).

Since the grasping tool is a simple parallel gripper, *EGA 1* and *2* will be successful only when the parent primitives individual orientations are orthogonal to the line connecting them, meaning that the two parent co-planar contours should be mirror symmetric and the two representative primitives should be positioned opposite to each other:

$$|\Theta_i \cdot \mathbf{D}| < C \quad \wedge \quad |\Theta_j \cdot \mathbf{D}| < C \quad (5)$$

where C is a positive real number smaller than one. If this is not the case, the grasp is unstable or not possible.

Both *EGA 3* and *4* give two grasping actions, one for every parent contour. *EGA 3* and *4* use the individual orientations of the parent primitives (projected to the common plane) as \mathbf{Y}_{TCP} direction and do not rely on the orientation of the connecting line. This is why orthogonality to the connecting line is not a requirement. The calculations are analog to the case of *EGA 1* (equation 5).

5. Experiments

This section gives a description of the experimental setup (section 5.1) and explains the testing procedure (section 5.2). Qualitative and quantitative results are given in section 5.3 and then become discussed in section 5.4.

5.1. Experimental setup

This section gives a description of the hardware (section 5.1.1) and software elements (section 5.1.2) used in the experimental setup.

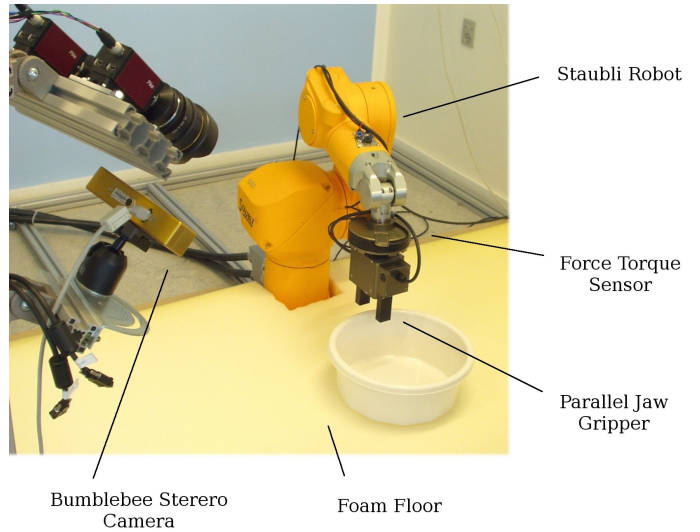


Fig. 7. Experimental setup.

5.1.1. Hardware

The hardware setup consists of a Staubli RX60 six degrees of freedom industrial robot arm, a fixed Bumblebee2 colour stereo camera, a FTACL 50-80 Schunk Force Torque sensor and a PowerCube 2-finger-parallel gripper tool mounted on the Force Torque sensor, (figure 7). The floor is covered with flexible foam layer. The stereo camera has a fixed position with respect to the robot. A common frame of reference is derived through a robot-camera calibration procedure.

The force torque sensor is used for active collision detection. The sensor is mounted between the wrist and the tool of the robot, and it measures forces or torques acting on the tool. By comparing forces and torques that can be expected from the influence of the gravitational force alone with those that are actually measured by the sensor, it is possible to detect any external collision or force that acts on the tool, (see figure 8).

The control application for executing the grasping attempts is run on a PC machine under Linux operating system. The system uses a Modbus interface to communicate to the Staubli robot and RS232 serial communication to communicate to the gripper and the force torque sensor. A firewire interface connects the camera to a Windows PC machine that exchange information with the control application through a TCP/IP connection.

5.1.2. Software

The implementation is based on three distinct software environments CoViS, RobWork [46] and Orocos [47]. CoViS is a cognitive vision system that is modelling early cognitive functions of biological visual systems, (section 3). It is being developed by the Cognitive Vision Group at University of Southern Denmark. RobWork is a framework for simulation and control of robotics with emphasis on industrial robotics and their applications. Orocos Real-Time Toolkit (RTT) is a C++ framework for implementation of

Difference Between Measured and Calculated Torque Total Over Time

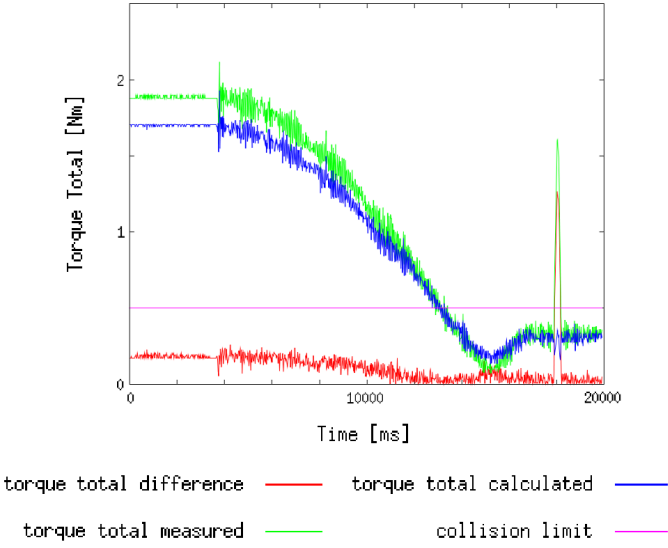


Fig. 8. The graph shows the total measured and the total calculated torques, and the difference between measured and calculated values as a function of time for a sample grasping attempt where a collision happened. Figure 12 shows an example collision situation and the corresponding grasping hypotheses.

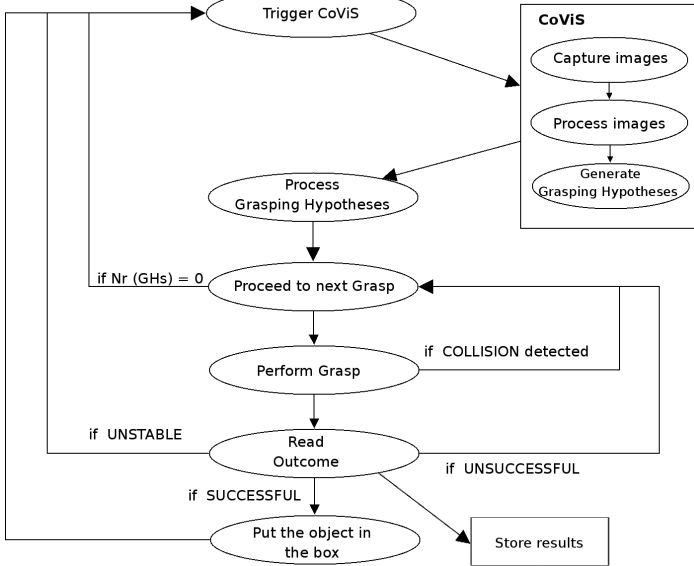


Fig. 9. State diagram showing workflow of exploration using the grasping behaviour.

(realtime and non-realtime) control systems. RobWork and Orocos are integrated into a single control application that communicates to the CoViS application using a TCP/IP connection.

5.2. Testing

Figure 9 shows the state diagram for the grasping procedure. The procedure starts by capturing and processing images, and producing grasping hypotheses (GHs). GHs

are then processed, certain number of grasping actions are tested and results are stored. The CoViS system creates many grasping hypotheses (from several thousand grasping hypotheses) for each scene, depending on the scene complexity and the quality of the reconstruction. As only few grasping hypotheses can be tested, (the scene will eventually become affected by robot actions), it is necessary to adopt a criteria for ranking the available hypotheses.

In this work, grasping hypotheses are first ranked by the amount of the verticality of the grasp, or more precisely:

$$R_S = Z_{TCP} \cdot (-Z_W)$$

where the ranking score R_S is in the interval $[-1,1]$. Z_{TCP} is the orientation of the Z axis of the TCP frame (see figure 6) expressed in the World reference frame, and $(-Z_W)$ is the vector pointing vertically down. Hence, grasps where the gripper fingers are pointing down vertically have the highest rank for eliminating such heuristics by learning see section 6.1.

The system then tries to find a maximum of five top ranked grasps that are reachable by the robot and can be accessed with a collision free movement of the robot. Collision free trajectories are calculated using the RRT-connect (Rapidly-exploring Random Trees) motion planner [48] with PQP (Proximity Query Package) collision detection strategy [49]. Figure 10d. shows the simulation environment used for motion planning. It includes the 3D models of the robot (kinematic and geometric) and the floor, and for each new scene it imports the reconstructed contours of the objects present in a region of interest in front of the robot.

Grasping attempts can result in *successful*, *unstable* or *unsuccessful* grasps or can report a *collision* in which case the robot stops and returns to the initial position. This evaluation is done autonomously by measuring the distance between the fingers. More precisely, we say that a grasp is

- *unsuccessful* if the distance between the fingers after closing (or opening) is 0 (or maximal),
- *unstable* if the distance is larger than 0 (or smaller than maximum) during the closing (opening) of the finger but 0 (or maximal) after having picked up the object,
- *successful* if the distance is larger than 0 (or smaller than maximum) during the closing (opening) of the finger as well as after picking up the object.

Moreover, *collisions* are detected by the force torque sensor.

5.3. Results

The experimental evaluation presented in this section is designed as an exploratory case analysis. The aim is to illustrate different aspects of the system's behaviour, its capabilities and weaknesses. Two types of experiments were performed.

In the first group of experiments (described in section 5.3.1), a test scene contains a single object. The robot attempts to remove it from the scene by using the grasping

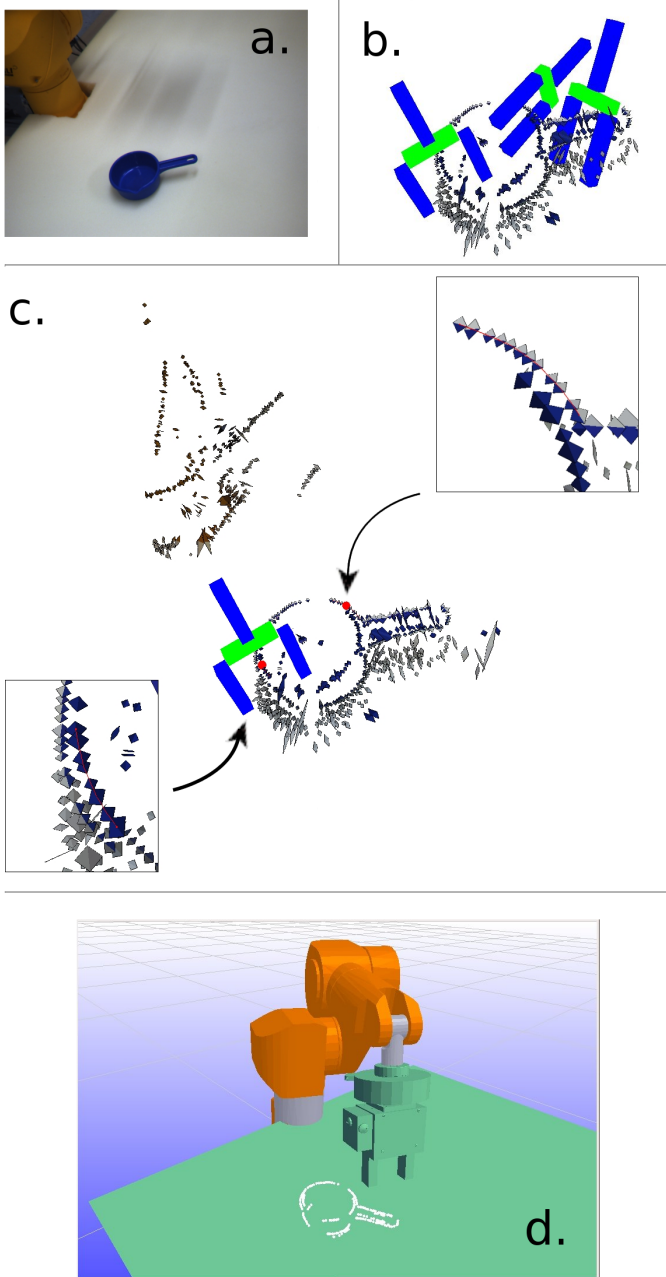


Fig. 10. a) The image taken during one of the experiments (Section 5.3.1) captured by the left camera. b) Some grasping hypotheses generated for that scene, displayed in a visualisation environment. c) A successful grasping hypotheses (EGA 3) where parent contours are magnified. The primitives in the top left corner come from the robot and the background. d) RobWork simulation environment shows 3D models of Staubli robot and floor. Additionally, the information about 3D edges in the scene is provided by the vision system. The 3D contours are composed of 3D primitives, which are modeled as small cubes. The models are used for planning collision free motions of the robot and for the visualisation purposes.

behaviour. Fourteen objects have been used in the evaluation (figure 11). The size and the shape of the objects are chosen so that grasping is connected to different degrees of difficulty.

The second group of experiments (described in section

Experimental situation	1	2
number of grasping hypotheses (GH)	66	373
number of accepted GHs	11	37
number of unreachable GHs	46	243
number of GHs where tool is in collision	9	93
number of GHs where collision free path was not found	0	0

Table 1

The results of processing full sets of GHs for the first two experimental situations (see figure 13). Finding a collision free path is an easy task due to the fact that the scene is not complex.

5.3.2) were performed on five complex scenes² containing a selection of the very same objects investigated in section 5.3.1 which are however distributed randomly with high degree of clutter and occlusion (see figure 15). The goal was to remove as many objects as possible from the scene. A short video showing an experimental setting similar to complex scenes described here is available at [44] (snapshots of the video are shown in figure 12).

5.3.1. Single objects

Each of the fourteen objects has been presented to the system in several different positions and orientations that vary in terms of grasping difficulty. Experiments performed with the first object are described in detail. Results on other experiments are given briefly.

Object 1

Three experiments were performed with object 1 (see figure 13). In the first experiment, the object was successfully grasped in the first attempt with the grasp of EGA 3 type. The same happened in the second experiment and the successful grasping hypotheses are shown on figure 10c. In the third experiment, the object was not grasped because it turned out to be unreachable by the robot. The object was also placed further away from the camera system than in the first two experiments, which gave a lower quality reconstruction and thus a fewer number of grasping hypotheses.

As mentioned in section 5.2, the ranked list of grasping hypotheses (GHs) is processed top-down. The processing stops when a certain number (five here) of accessible GHs have been found, or when there are no more GHs available (see section 4.1). In order to give an illustration of a typical processing outcome, full sets of GHs have been processed for the first two experimental situations, and the results are shown in table 1. The order of the conditions that a grasping hypothesis has to fulfill in order to be accepted is identical to the order in the table, (e.g., GHs are first checked for reachability, then it is checked whether the position of the tool during grasping is collision free and if both of those conditions are satisfied the system will try to calculate a collision free trajectory).

² We show results on three of these scenes. Results on the other two scenes are described in [50]

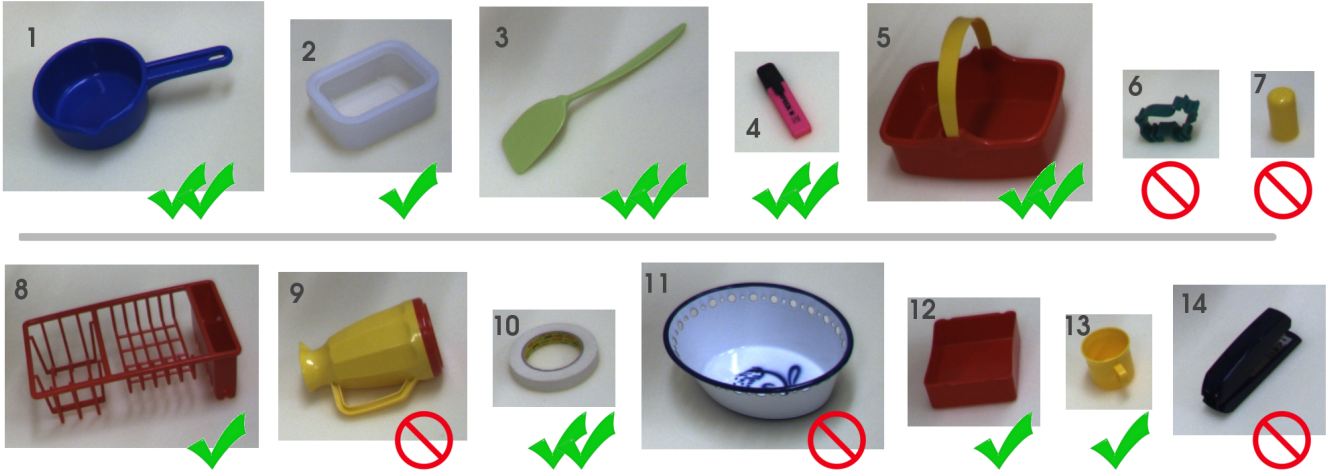


Fig. 11. Office and toy kitchen objects used in the experiments. Objects are of mostly uniform colours, and their size and the shape is suitable for grasping with the parallel jaw gripper. The marks illustrate average success rate in grasping individual objects measured in the experiments (see also table 3).

Experimental situation	1	2
number of contours	27	30
number of all contours pairs	351	435
number of similar contours pairs	201	241
number of accepted parents pairs	17	94
number of discarded parents	184	147
number of GHs	66	373

Table 2
Intermediate values from grasping hypotheses generation program.

As can be seen from table 1, only a small percentage of the computed grasping hypotheses become actually performed. Most computed grasping hypotheses can be disregarded by constraints that can be computed beforehand.

Table 2 shows some intermediate values from the grasping hypotheses generation program for the first two scenes of figure 13. The number of contours, contour pairs and the number of similar contour pairs are derived from the whole image representation. Parent primitive pairs are then assigned to the pairs of similar contours. A parent pair is discarded if any of the two primitives does not belong to a certain region of interest in front of the robot. Background features that originate from the robot and the edge of the ground surface (figure 13) generate a lot of undesirable similar contours and that is why the number of discarded parent pairs is high. This however does not explain why there is a significant difference between number of good parent pairs and consequently generated grasping hypotheses in the two cases. The difference arises because the representation of object 1 contains less detail in the first case, as it is further away from the camera.

As mentioned in the introduction of this section, it is important to notice that individual experimental situations were designed to demonstrate different aspects of the system's performance and are not suitable for direct statis-

object nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
successful grasps (%)	67	30	50	50	50	0	0	25	0	50	0	33	17	6
unstable grasps (%)	0	0	3	0	0	0	0	18	0	0	100	33	0	11
collisions (%)	0	42.5	16	0	39	33	17	41	77	11	0	0	29	83
unsuccess grasps (%)	33	27.5	31	0	11	0	33	16	23	39	0	33	29	0
no grasps (%)	0	0	0	50	0	66	50	0	0	0	0	0	25	0

Table 3
The results of experiments with single objects.

tical analysis. However, we still present a weak numerical comparison of the experimental results on different objects. Figure 14 shows experimental situations for the 14 objects. Table 3 gives the corresponding distribution of different grasping outcomes.

One of the factors that influences the outcome of a grasping attempt is the placement of the object with respect to the camera since reconstructed primitives have uncertainties that vary with the distance from the image centre and with the distance from the camera. Small objects that are placed too far away also do not have a good enough reconstruction for triggering grasps. Object 11 turned out to be too heavy to be lifted from the ground. Objects 3, 4, and 10 have edges that are positioned very close to the floor so that small errors in the vertical direction can cause collisions with the floor. In some cases (object 12 - situation 3, object 5 - situation 3, object 2 - situation 2) the object's opening was not available for vertical (top down grasps) which are ranked highest, so that potentially successful grasps with non-vertical orientations were not chosen. In few cases shadows triggered grasping attempts.

5.3.2. Multiple objects

In the second evaluation stage, grasping hypotheses were tested on three complex scenes. For each scene, the robot performed 30 grasping actions in order to remove as many



Fig. 14. Experimental situations for objects 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14. Photos captured by the left camera.

objects as possible from a scene.

Figure 15 show three complex scenes. Photos on the left show initial situations and photos on the right show the same scenes after performing 30 grasping attempts. As can be seen by comparing the changes, even in these complex scenes with many objects, strong occlusions and clutter, a good number of grasping attempts have been successfully performed.

Table 4 gives results of the experiments for complex scenes. The relative success of the grasping behaviour depends on the number of the attempts taken into account. The 30 grasping attempts were usually enough for the system to perform all possible successful grasps. We experienced that in case the system continues working after this point, the number of the unsuccessful, collision and unsta-

random scene	1	2	3
number of grasping attempts	30	30	30
successful grasps	6	4	5
unstable grasps	5	2	3
collisions	18	12	16
unsuccessful grasps	1	12	6

Table 4

The results of experiments with complex scenes 1, 2 and 3.

ble outcomes grows. It happens because the remaining objects are not in the suitable position, (unreachable or graspable edges are not visible), or due to the ranking criteria, some nonsuccessful grasps repeatedly become favoured so that other possibilities are not explored.

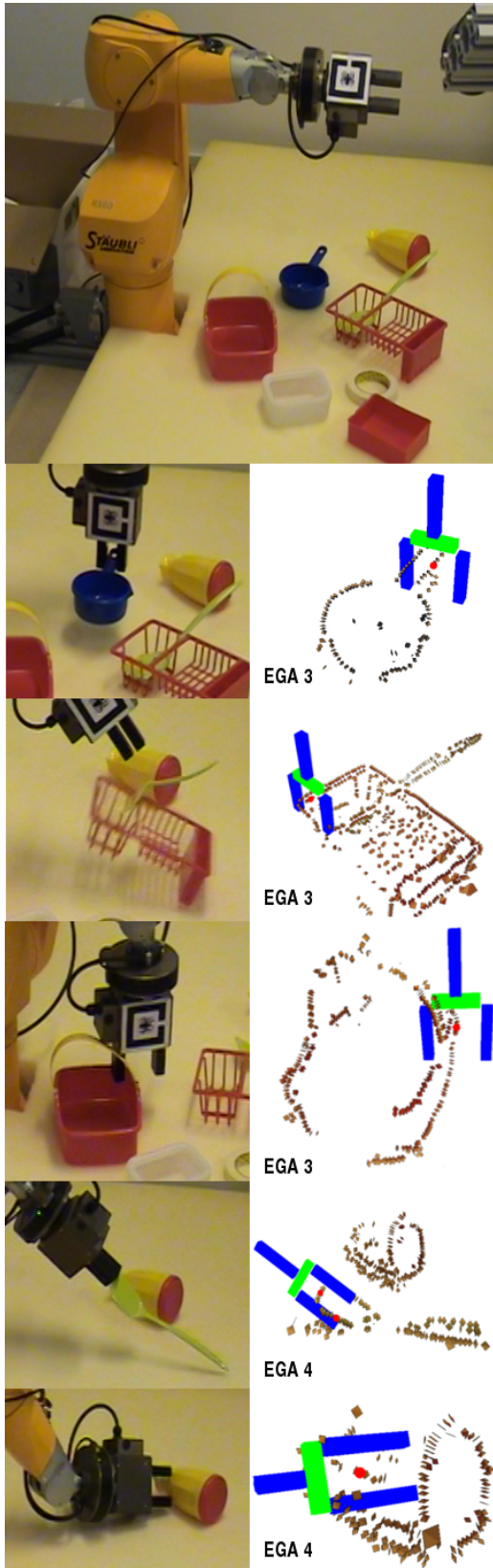


Fig. 12. Five grasping outcomes from the video available at [44]. From top to bottom: Successful, Unstable, Collision, Successful and Fail cases. A snapshot from the video (left) and the corresponding grasping hypotheses viewed in a visualisation environment (right) is shown for each case.

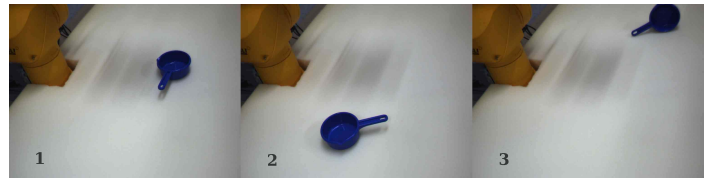


Fig. 13. Three experimental situations for the object 1. Figure shows the original images used for acquiring image representations, captured by the left camera. The darker areas at the middle of all three images are shadow cast by the robot when in the initial position.



Fig. 15. Complex scenes 1, 2 and 3. Images on the left show initial scenes. Images on the right show the corresponding scenes after 30 grasping attempts with our grasping behaviour.

In a complex scene, grasping hypotheses can be defined with edges from two different objects. The use of the co-colourity relation (i.e., two primitives sides facing each other have the same colour, see section 3.3) make it likely that the parent primitives are from the same object. However, the outer colour of edges of the two objects is usually the colour of the floor surface and if the two edges are co-planar at the same time, a grasping hypotheses will be created. In most cases, this is not a disadvantage as GHs originating from different objects often give good results.

5.4. Discussion of Experiments

The grasping experiments performed on single objects as well as those performed on cluttered scenes with many objects showed that there is a consistency in graspability of specific objects. In other words, some objects are grasped

	percentage
EGA 1	10%
EGA 2	5%
EGA 3	50%
EGA 4	35%

Table 5
Distribution of EGA types for successful grasps in single objects experiments.

easily and consistently whenever they are in suitable position and image processing produces a good representation. Other objects are grasped just occasionally. This depends on how well individual object’s features (weight, size, shape, colour, material) pair with the type of gripper used in the experiments. On the other hand, it depends on how suitable the object’s features are for the kind of image processing used, i.e. how difficult it is to extract good co-colour and co-planar contours. For small or distant objects, the reconstruction was often poor. In these cases, images with higher resolution or making use of a visual attention mechanism could improve the performance.

The gripper used in this setup limits grasps of EGA 1 and EGA 2 types only to small objects. Large objects are mostly grasped by the edges with grasps of type EGA 3, if they are concave. Although object 9 could be grasped by the handle, this did not happen because the algorithm does not identify the handle as a specifically good grasping position. Here object dependent grasping knowledge (see section 6.2) acquired by supervised learning (e.g. by imitation learning (see, e.g., [25])) might become an important option for improvement.

Table 5 gives the distribution of EGA grasp types for the successfully performed grasps in the experiments with single objects. EGA types 3 and 4 are represented more often because EGA types 1 and 2 have two additional constraints, (the distance between grasped edges has to be within the opening range of the gripper, and the edges have to be mirror symmetric).

The current system has an open loop - ”look-and-move” type of control. The drawback of this is the high sensitivity to calibration errors. The accuracy of grasping operation depends directly on the accuracy of the visual sensor, the robot end-effector, and the robot-camera calibration. This could be avoided by visual servoing. However, in our calibrated set up this was not necessarily required.

The exploration procedure could be additionally enhanced by making use of tactile sensors and based on that, reactive grasping strategies (see, e.g., [41]). Our simple grasping strategy could then serve as an initial ”approach” planner. This would potentially reduce the number of unstable grasps and would also give rich feedback for learning (see also section 6).

6. Grasping Reflex in a Cognitive System

The grasping behaviour introduced in this paper is an important part of the cognitive system developed with the project PACO-PLUS [51]. Of particular importance is that the success of the action can be evaluated autonomously by the system. In our case, haptic information from the gripper can be used to distinguish between successful, unstable and unsuccessful grasps as described in section 5.2. Hence, some kind of an episodic memory (see, e.g., [52]) can be build up autonomously that can then be used for further refinement of the grasping behaviour. In that context, we have defined a learning procedure that allows for improving the grasping behaviour by making use of the grasping attempts and their evaluation stored in the episodic memory as described in section 6.1. Moreover, by means of the grasping behaviour defined in this paper, we are able to build up world knowledge in terms of object representations and associated grasps (as described in section 6.2).

6.1. Refinement of Grasping Reflex

The exploration behaviour described in this paper performs multiple autonomously evaluated grasping attempts, stored in the episodic memory. This memory can be used as input for learning since it preserves information that gives indications about likelihoods of success or failure. For example, if the parent primitives are more distant than the maximal distance between fingers allowed by the gripper, EGA 1 and 2 are not executable anymore. Another example is the uncertainty of the reconstruction of the primitives that depends on the distance of the object to the camera as well as the ’eccentricity’ (i.e., the distance to the principal ray of the camera) of the parent primitives (for an exact analysis of the uncertainty see [53]). Hence, it is possible to learn the relation between these parameters and the success likelihood of a grasping attempt.

More specifically, we have used the such parameters extracted from the evaluated grasping attempts as a basis for learning algorithm. A grasp is associated with two parent primitives, Π_i and Π_j . The 3D positions of those are denoted X_i and X_j . The position of the grasp is denoted P_{TCP} . C_L and C_R denote the positions of the optical centre of the left resp. right camera. The following features, illustrated in figure 16, have been computed:

- F1 The height, h . It is given by the z-component of P_{TCP} .
- F2 The angle, φ_v , between the normal, n_p , (equation 4) and a vector in the world reference frame pointing vertically up
- F3 The distance between parent primitives, d_p .
- F4 The distance to the camera, d_{cam} .
- F5 The angle, φ_{cam} , defined by a ray from the optical center to TCP and vector pointing normally out of the image plane.

F1–F3 refer to the robots ability to grasp the object independently of where the object is positioned. F4 and F5

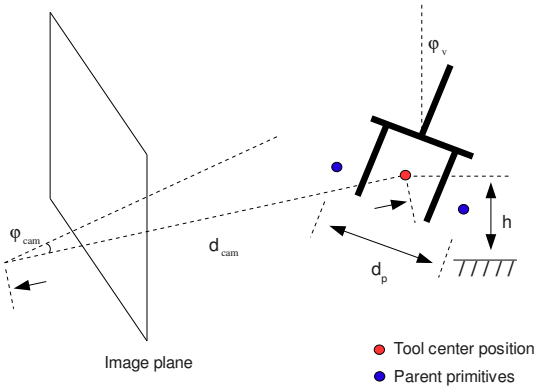


Fig. 16. Features used for learning.

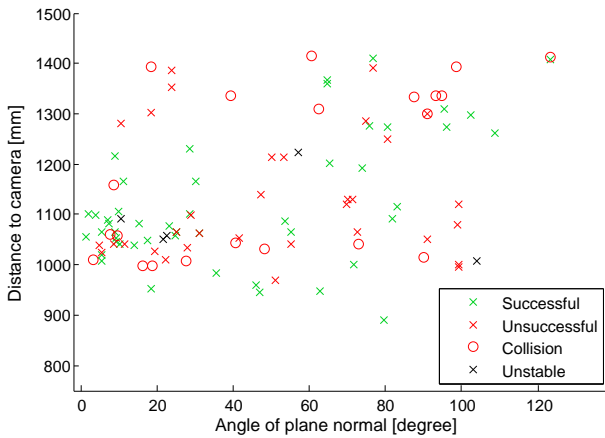


Fig. 17. Distribution of recorded grasps. Unstable grasps have not been used for learning.

are related to the relative position of camera and object which are the reason for large variation of the quality of the 3D reconstruction which will have an effect on the quality of the computed EGA. The grasping attempts used for learning are shown in figure 17 with respect to φ_{cam} , d_{cam} and the outcome of the grasps. It can be observed that the success of the grasp (indicated as a green cross) depends on F4 and F5. All features can be computed using the 3D positions of the parent primitives and the camera calibration parameters. In addition, some EGAs might be more robust to reconstruction errors or wrong interpretations of data. Therefore the type of the grasp needs to be taken into consideration as well. The learning is implemented using a radial basis function network (for details of the structure of the network as well as for a detailed analysis of the features F1–F5 we refer to [54]).

The effect of the learning is tested by randomly dividing the data set stored in the episodic memory into training set and a test set, containing about 117 resp. 20 evaluated grasps. Then the 10 grasps which get the highest score from the network are selected from the test set. The success ratio of these is compared to the success ratio of 10 grasps that have been selected randomly from the same test set. This procedure has been repeated 20 times. The average performance could be increased from below 35% to more

than 45%.

6.2. Building-up world knowledge

Grasping objects in unconstrained environments without any specific prior object knowledge is known to be a very difficult problem as outlined in the section 1. Hence, a performance close to 100% is not to be expected. Although humans can solve this problem, it needs to be acknowledged that this skill only develops after years of learning (see, e.g., [55]) and hence is likely to make use of a vast amount of experience with a variety of objects. However, once the object is known to the system, a much higher performance is achievable. The grasping behaviour described in this paper has been used to generate such object knowledge and to learn grasping based on this knowledge, i.e., to build up general world knowledge by learning, as described now.

The early cognitive vision system described in section 3 is able to extract 3D representations (see figure 18) of objects by accumulating information over different frames (see [56]). A pre-requisite for using this accumulation mechanism is that the robot has physical control over the object (see figure 18a), allowing the robot to perform movements leading to predictable transformations of the 3D primitives. Based on these predictions, Kalman filtering can be used to refine the estimates of 3D features and Bayesian reasoning can be used to eliminate wrong 3D features, leading to reliable 3D object representations (see figure 18b). The physical control is achieved by means of the grasping behaviour described in this paper. Although not leading to a close to 100% success rate, by means of our haptic evaluation, the system is able to judge whether something has been grasped successfully. Then the robot can perform a controlled movement (mostly a rotation) of the object, that can be used in the accumulation algorithm to extract complete and reliable object representations as shown in figure 18b).

Moreover, the computation of grasping hypotheses only requires a co-planarity and co-colourity relation. The co-planarity relation as well as the co-colourity relation are also defined for these accumulated representations which also consist of 3D primitives. Hence, also grasping hypotheses can also be computed for such stored models (see figure 18c) and be tested after a successful pose estimation has been performed (as done, in e.g., [57,58] based on the accumulated representations). This mechanism has been used in the PACOplus system to learn grasp densities associated to a specific object (see figure 18d and [14]). By that, the grasping mechanism introduced in this paper which does not require any object prior knowledge has been used to bootstrap a system which generates an object specific grasping mechanism with a higher success rate due to the larger prior being incorporated.

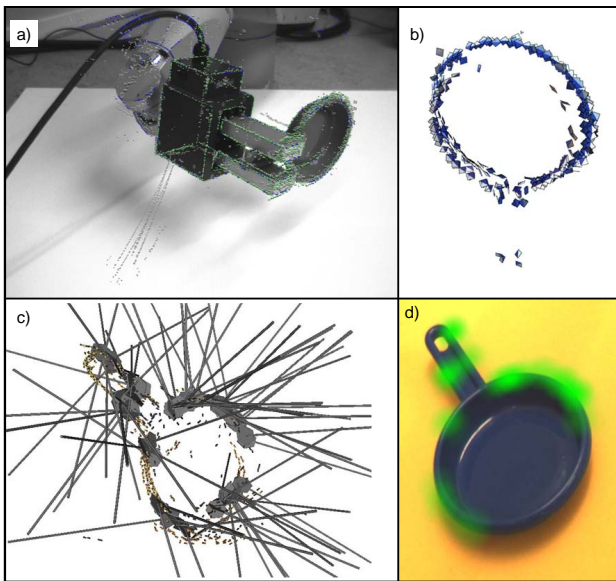


Fig. 18. a) An image of the object held by the robot where tracked 3D primitives are indicated as local line segments. b) Illustration of the object model consisting of successfully tracked 3D-primitives. Note that the hole at the handle originates from the fact that the gripper holds the object at the handle and hence the handle is occluded for the vision system. c) Grasping hypotheses generated by our algorithm extracted based on the learned representation shown in b). d) Projection of grasp densities (note that although the grasping density is a 6D manifold, only 3D positions are shown) extracted from empirically tested grasping hypotheses found being successful.

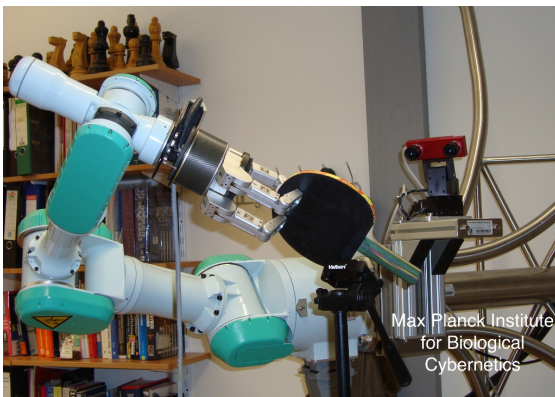


Fig. 19. Execution of a grasping hypothesis with a three finger hand in the context of the work in [14].

7. Summary and conclusion

We have described a grasping mechanism that does not make use of any specific object prior knowledge. The mechanism makes use of second order relations between visually extracted 3D features representing edge structures. We showed that our algorithm, although making use of such rather simple constraints, is able to grasp objects with a reasonable success rate in rather complex environments. Meanwhile, the grasping mechanism has also been used in a system with a different embodiment as shown figure 19.

Moreover, we have described the role of our grasping be-

haviour within a cognitive system. The system is able to evaluate the success of the grasps autonomously by haptic feedback. By this it can create ground truth in terms of labelled data that has been used for improving the initially hard-wired algorithm by learning. Moreover, the grasping behaviour has been used to trigger higher level processes such as object learning and learning of object specific grasping.

Grasping without prior object knowledge is a task in which multiple cues need to be merged. In this way, we see our 3D approach as complementary to other mechanism based on 2D information (such as, e.g., [59,10]) or 3D surface information (such as, e.g., [18]).

8. Acknowledgements

This work has been funded within the PACOplus project (IST-FP6-IP-027657). We would like to thank Oliver Kroemer for providing figure 19 and Renaud Detry for providing figure 18c) and d). We also thank Morten Kjaergard for ground work on the OROCOS control application.

References

- [1] A. Bicchi, V. Kumar, Robotic grasping and contact: A review, in: Proceedings of IEEE International Conference on Robotics and Automation, 2000, pp. 348–353.
- [2] L. Natale, F. Orabona, G. Metta, G. Sandini, Exploring the world through grasping: a developmental approach, Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on (2005) 559–565.
- [3] G. Recatalá, E. Chinellato, A. P. D. Pobil, Y. Mezouar, P. Martinet, Biologically-inspired 3D grasp synthesis based on visual exploration, Autonomous Robots 25 (1-2) (2008) 59–70.
- [4] K. Huebner, S. Ruthotto, D. Kragic, Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping, in: Proceedings of the 2008 IEEE International Conference on Robotics and Automation, 2008, pp. 1628–1633.
- [5] C. Borst, M. Fischer, G. Hirzinger, Grasp Planning: How to Choose a Suitable Task Wrench Space, New Orleans, LA, USA, 2004, pp. 319 – 325.
- [6] D. Ding, Y.-H. Liu, S. Wang, The synthesis of 3-d form-closure grasps, Robotica 18 (1) (2000) 51–58.
- [7] M. Buss, H. Hashimoto, J. B. Moore, Dextrous hand grasping force optimization, Robotics and Automation, IEEE Transactions on 12 (3) (1996) 406–418.
- [8] R. Platt, Learning Grasp Strategies Composed of Contact Relative Motions, in: IEEE-RAS International Conference on Humanoid Robots, 2007.
- [9] R. Pelosof, A. Miller, P. Allen, T. Jebara, An SVM Learning Approach to Robotic Grasping, Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on (2004) 3512–3518 Vol.4.
- [10] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, A. Y. Ng., Learning to grasp novel objects using vision, In 10th International Symposium of Experimental Robotics (ISER).
- [11] N. Krüger, M. Lappe, F. Wörgötter, Biologically Motivated Multi-modal Processing of Visual Primitives, The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour 1 (5) (2004) 417–428.

- [12] N. Pugeault, Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation, VDM Verlag Dr. Müller, 2008.
- [13] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragic, S. Kalkan, F. Wörgötter, N. Krüger, Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes, Special Issue on "Cognitive Humanoid Robots" of the International Journal of Humanoid Robotics(accepted).
- [14] R. Detry, M. Popovic, Y. P. Touati, E. Baseski, N. Krüger, J. Piater, Autonomuous learning of object-specific grasp affordance densities, submitted to the 8th International Conference on Development and Learning.
- [15] A. Bicchi, On the closure properties of robotic grasping, International Journal of Robotics Research 14 (1995) 319–334.
- [16] H. Maekawa, K. Tanie, K. Komoriya, Tactile feedback for multifingered dynamic grasping, Control Systems Magazine, IEEE 17 (1) (1997) 63–71.
- [17] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, J. Wikander, Demonstration based Learning and Control for Automatic Grasping, in: Proc. of the International Conference on Advanced Robotics, 2007.
- [18] A. T. Miller, S. Knoop, H. Christensen, P. K. Allen, Automatic grasp planning using shape primitives, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2003, Vol. 2, 2003, pp. 1824–1829.
- [19] K. K. Aydin, Fuzzy logic, grasp preshaping for robot hands, in: ISUMA '95: Proceedings of the 3rd International Symposium on Uncertainty Modelling and Analysis, IEEE Computer Society, Washington, DC, USA, 1995, pp. 520 – 523.
- [20] M. R. Cutkosky, On grasp choice, grasp models, and the design of hands for manufacturing tasks, IEEE Transactions on Robotics and Automation 5 (3) (1989) 269–279.
- [21] T. Iberall, Human prehension and dexterous robot hands, The International Journal of Robotics Research 16 (3) (1997) 285–299.
- [22] A. T. Miller, P. K. Allen, GraspIt!: A Versatile Simulator for Robotic Grasping, Robotics & Automation Magazine, IEEE 11 (4) (2004) 110–122.
- [23] J. Jørgensen, H. Petersen, Usage of simulations to plan stable grasping of unknown objects with a 3-fingered Schunk hand, in: Workshop on robot simulators: available software, scientific applications and future trends, ICRA, 2008.
- [24] S. Ekvall, D. Kragic, Integrating object and grasp recognition for dynamic scene interpretation, in: IEEE International Conference on Advanced Robotics, 2005. ICAR'05, 2005, pp. 331–336.
- [25] J. Steil, F. Röthling, R. Haschke, H. Ritter, Situated robot learning for multi-modal instruction and imitation of grasping, Robotics and Autonomous Systems Special Is (47) (2004) 129–141.
- [26] R. Dillmann, M. Kaiser, A. Ude, Acquisition of elementary robot skills from human demonstration, in: In International Symposium on Intelligent Robotics Systems, 1995, pp. 185–192.
- [27] Scape Technologies, <http://www.scapetechnologies.com/>.
- [28] M. Salganicoff, L. H. Ungar, R. Bajcsy, Active learning for vision-based robot grasping, Machine Learning 23 (2-3) (1996) 251–278.
- [29] R. Mario, V. Markus, Grasping of unknown objects from a table top, in: Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments,ViA08, 2008.
- [30] M. J. Taylor, A. Blake, Grasping the Apparent Contour, in: ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision, Springer-Verlag, London, UK, 1994, pp. 25–34.
- [31] G. Bekey, H. Liu, R. Tomović, W. Karplus, Knowledge-Based Control of Grasping in Robot Hands Using Heuristics from Human Motor Skills, IEEE Trans. Robotics and Automation vol. 9, no. 6 (1993) 709–722.
- [32] E. Chinellato, R. B. Fisher, A. Morales, A. P. D. Pobil, Ranking planar grasp configurations for a three-finger hand., in: ICRA, 2003, pp. 1133–1138.
- [33] A. Morales, P. J. Sanz, A. P. D. Pobil, A. H. Fagg, Vision-based three-finger grasp synthesis constrained by hand geometry, Robotics and Autonomous Systems 54 (6) (2006) 496–512.
- [34] D. P. Perrin, C. E. Smith, O. Masoud, N. Papanikolopoulos, Unknown Object Grasping Using Statistical Pressure Models, in: Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA, 2000, pp. 1054–1059.
- [35] G. Taylor, L. Kleeman, Grasping unknown objects with a humanoid Robot, Proceedings 2002 Australasian Conference on Robotics and Automation (2002) 191–196.
- [36] F. Ade, M. Rutishauser, M. Trobina, Grasping unknown objects, in: Proceedings of Dagstuhl Seminar: Environment Modeling and Motion Planning for Autonomous Robots, World, 1995, pp. 445–459.
- [37] B. Wang, L. Jiang, J. LI, H. Cai, Grasping unknown objects based on 3D model reconstruction, Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on (2005) 461 – 466.
- [38] J. A. C. Jr., J. H. Piater, R. A. Grupen, Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot, Robotics and Autonomous Systems 37 (2-3) (2001) 195–218.
- [39] J. J. Gibson, The Ecological Approach to Visual Perception, Lawrence Erlbaum Associates.
- [40] G. Fritz, L. Paletta, M. Kumar, G. Dorffner, R. Breithaupt, E. Rome, Visual Learning of Affordance Based Cues, in: Simulation of Adaptive Behavior, Vol. 4095, 2006, pp. 52–64.
- [41] J. Steffen, R. Haschke, H. Ritter, Experience-based and Tactile-driven Dynamic Grasp Control, Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on (2007) 2938–2943.
- [42] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, J. Wikander, Demonstration based learning and control for automatic grasping, in: Int. Conf. on Advanced Robotics, Jeju, Korea, 2007.
- [43] D. Aarno, J. Sommerfeld, D. Kragić, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, N. Krüger, Early Reactive Grasping with Second Order 3D Feature Relations, in: IEEE International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision, 2007.
- [44] Grasping Video, <http://www.mip.sdu.dk/covig/videos/graspingReflexCompressed.divx>.
- [45] N. Pugeault, F. Wörgötter, N. Krüger, Multi-modal Scene Reconstruction Using Perceptual Grouping Constraints, in: Proc. IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06), 2006.
- [46] RobWork, <http://www.mip.sdu.dk/robwork/>.
- [47] The Orocos Real-Time Toolkit, <http://www.orocos.org/rtt>.
- [48] J. J. Kuffner, J. Steven, M. Lavalley, Rrt-connect: An efficient approach to single-query path planning, in: In Proc. IEEE Intl Conf. on Robotics and Automation, 2000, pp. 995–1001.
- [49] E. Larsen, S. Gottshalck, M. Lin, D. Manocha, Fast proximity queries with swept sphere volumes, Tech. Rep. TR99-018, Department of Computer Science, University of North Carolina (1999.).
- [50] M. Popović, An Early Grasping Reflex in a Cognitive Robot Vision System, Master's thesis, Cognitive Vision Lab, The Mærsk Mc-Kinney Møller Institute, University of Southern Denmark (2008).
- [51] PACO-PLUS: Perception, Action and Cognition through learning of Object-Action Complexes, iST-FP6-IP-027657, Integrated Project (2006-2010).
- [52] A. D. Baddeley, Essentials of Human Memory, Psychology Press, Taylor and Francis, 1999.

- [53] N. Pugeault, S. Kalkan, E. Başeski, F. Wörgötter, N. Krüger, Reconstruction uncertainty and 3D relations, in: Proceedings of Int. Conf. on Computer Vision Theory and Applications (VISAPP'08), 2008.
- [54] L. Bodenhausen, Project in Artificial Intelligence (2008).
- [55] P. J. Kellman, M. E. Arterberry, The Cradle of Knowledge: Development of Perception in Infancy (Learning, Development, and Conceptual Change), The MIT Press, 1998.
- [56] N. Pugeault, F. Wörgötter, N. Krüger, Accumulated visual representation for cognitive vision, in: British Machine Vision Conference, 2008.
- [57] R. Detry, N. Pugeault, J. H. Piater, Probabilistic Pose Recovery Using Learned Hierarchical Object Models, in: International Cognitive Vision Workshop (Workshop at the 6th International Conference on Vision Systems), 2008.
- [58] R. Detry, J. Piater, A probabilistic framework for 3d visual object representation, IEEE PAMI.
- [59] J. Pauli, H. Hexmoor, M. Mataric, Learning to recognize and grasp objects, Autonomous Robots 5 (1998) 239–258.

Computational Vision and Active Perception
Centre for Autonomous Systems
School of Computer Science and Communication
KTH, Stockholm, 10044, Sweden
celle@nada.kth.se, babak2@kth.se, khubner@kth.se, danik@kth.se

An active vision system for detecting, fixating and manipulating objects in real world

Babak Rasolzadeh, Mårten Björkman, Kai Hübner, Danica Kragic

Abstract

The ability to autonomously acquire new knowledge through interaction with the environment is one of the major research goals in the field of robotics. The knowledge can be acquired only if suitable perception-action capabilities are present. In other words, a robotic system has to be able to detect, attend to and manipulate objects in the environment. 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 discussed. More precisely, we present a stereo based vision system framework where aspects of Top-down and Bottom-up attention and foveated attention are put into focus and demonstrate how the system can be utilized for 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). In addition, as pointed out in [74], a significant amount of human visual processing is not accessible to consciousness - we do not *experience* using optical flow to control our posture. 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 approach.

In our service robot project, tasks such as "Robot, bring me my cup" or "Robot, pick up this" are studied. Depending on the task or context information, different strategies 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, however it knows its approximate *location*. A different set of underlying visual strategies are required for each of these scenarios which are the most representative examples for robotic fetch-and-carry tasks.

We have been working with different aspects of the above for the past several years, [60, 79, 4, 19, 18, 42, 20]. The work presented here is in the line with these previous works, with the slight difference that we keep our focus to the design and development of a vision system architecture that allows for more general solutions in the above settings and suitable also for object manipulation. The goal of the designed system is to enable a generic object finding and manipulating platform. Although the specific robotic platform that we have implemented this design on is fixed (i.e. attached to the floor), the reasoning (and the design) is not limited to a static robotic platform. The reasoning can easily be extended to a service robot scenario, which is the long-term purpose and goal of our research.

A general overview of the robotic platform is given in the schematic illustration of Fig. 1. The different parts of this illustration will be described in detail throughout this paper.

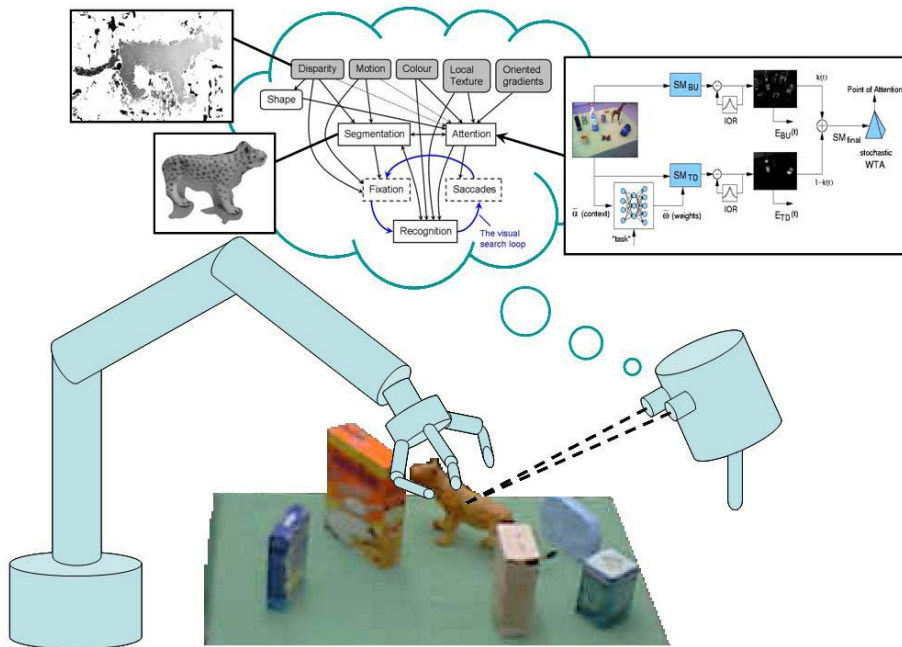


Fig. 1: Illustration of the complete robotic platform that is the system described in this paper.

The rest of 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 (the robot-head in the same illustration) and its calibration are given. Aspects of Bottom-up and Top-down attention (the Attention-box to the right in the illustration) are presented in Section 4 and foveated segmentation in Section 5 (the Segmentation-box to the left in the illustration). 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 (Section 7.1) and the recognition-system (Section 7.2) is done separately first, followed by a find-and-grasp task for the robot 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 larger 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 can not at all be expected?

Recent works exhibiting this sort of embodiment of vision are the work of Ude et al. [1], as well as the work of Björkman & Eklundh [6]. 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.

For the visual system to be of use to the robotic system, it needs the abilities to divide the world into *things*, create representations of observed *things* for later association and manipulation, and continuously update these 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 [82, 83, 84]. 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. We call such a mechanism 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. More on attention 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 its doings. 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 work presented here does not include the motion path, but more information can instead be found in [5].

The above-mentioned vision system has been implemented on the four-

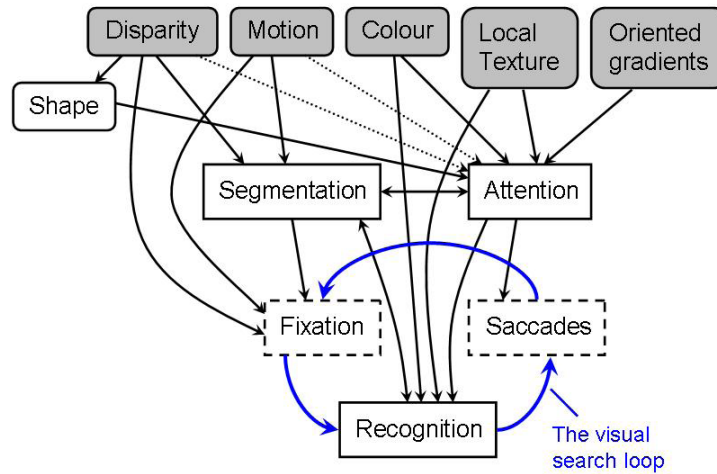


Fig. 2: The flow of visual information.

camera stereo head [77] shown in Fig. 3. The head consists 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 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.

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, biological plausibility comes at a cost. One critical difference is the relative costs of computation and communication of computed 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 tends 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 outweighs the cost of computation. Thus for a system to perform at real-time speed, biological plausibility easily becomes a hindrance. Even if



Fig. 3: The Armar-III stereo head.

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 [69, 43] that uses 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 [85, 59]. 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.

There is nothing 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 [70], a wide-field set for attention and a foveal one for recognition and manipulation. The most important dis-

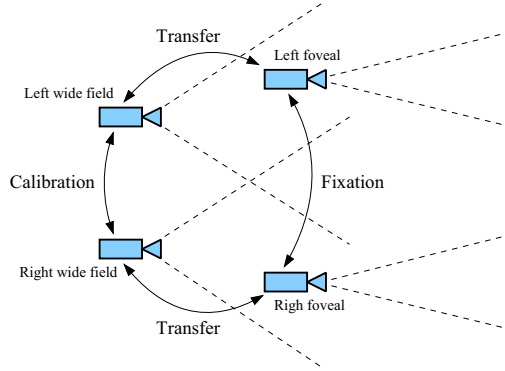


Fig. 4: Two sets of cameras, a wide-field camera set for attention and a foveal one for recognition and manipulation, with external calibration performed between pairs.

advantage 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 [28], 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 pair-wise camera calibration as illustrated by the arrows in Fig. 4.

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 following frames, assuming a limited change in relative camera orientation from one frame to the next. For feature matching we use Harris' corner features [26] and random sampling [22]. The wide-field cameras are related to each others using an iterative approach based on the optical flow model [46]

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

In an earlier study [3] 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 [47].

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 such that the object is placed in fixation in the foveal views. This is done using affine transfer [21], 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 be 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 $\mathbf{x}_w^\top \mathbf{F}_A \mathbf{x}_f = 0$ and the affine essential matrix

$$\mathbf{F}_A = \begin{pmatrix} 0 & 0 & a_3 \\ 0 & 0 & a_4 \\ a_1 & a_2 & a_5 \end{pmatrix}. \quad (2)$$

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. With the wide-field views related using Equation (1) and the foveal views related to their wide-field counterparts using Equation (2), a new fixation point can be transferred from the wide-field to the foveal views.

3.3 Fixation

Once a transfer has been completed and a saccade (a rapid shift in view point) 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 [27] are extracted from both views and features are matched using random sampling [22]. 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 our robotic system is as a selection mechanisms serving the higher level tasks such as object recognition and manipulation. Biological systems provide a good basis for these kinds of studies. 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 do 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) [34, 45]. 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 [58]. 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 [38], given these sources of information. 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 be governed by a differential equation that can be solved at least numerically for some special cases. 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 [80]. In computer vision, attentive processing for scene analysis initially largely dealt with salience based models, following [80] and the influential model of Koch and Ullman [38]. 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 [39] propose that a stochastic 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 [63] calculate feature and background statistics to be used in a game theoretic WTA framework for detection of objects. Choi et al. [11] suggest learning the desired modulations of the saliency map (based on the Itti and Koch model [36]) for Top-down tuning of attention, with the aid of an ART-network. Navalpakkam and Itti [55] 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. [44] 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 [23] 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. [57] 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.

One shortcoming of most of these computational models is that they are usually limited to the study of attention itself, and other than some works on the use of attention for object recognition, it has never been studied in a broader visual system perspective such as a service robotic context. One of the few recent works that does in fact incorporate an computational mechanism for attention into a humanoid platform is the work of Morén et al. [52] They use a method called Feature Gating to achieve Top-down modulation of saliencies.

Our framework is based on the notion of saliency maps, SMs [38]. 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 kinds 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 ones. The model (illustrated in Fig. 5) 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 (the Neural Network). An Inhibition-of-Return mechanism and stochastic Winner-Take-All network gives the system its dynamic behavior described in Section 4. 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.

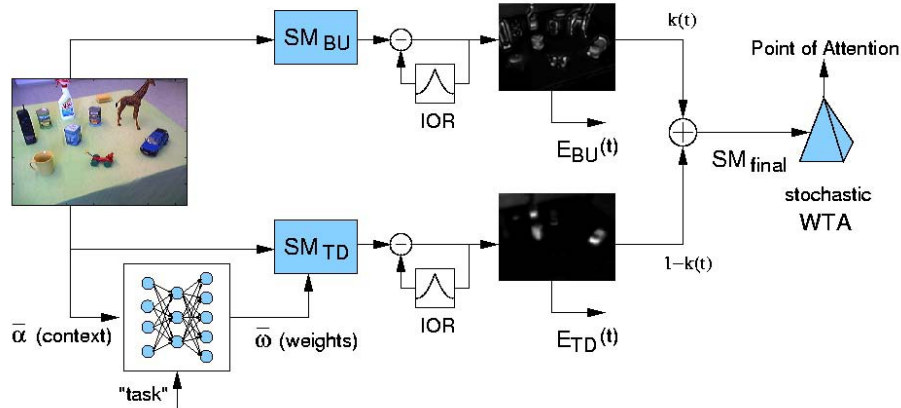


Fig. 5: An attentional model that combines Bottom-up and Top-down saliency, with Inhibition-of-Return and a stochastic Winner-Take-All mechanism, with context and task dependent Top-down weights.

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 Koch-Ullman model [38] that was (and is) developed by the group around Itti et al. [36, 35, 55]. 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 [16]. 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 [78, 67].

4.2 Weight optimization and contextual learning

As mentioned above we base both Top-down and Bottom-up saliency 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 Frintrap [23], the weighting is done differently. This has important consequences, as 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. 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:

$$\begin{aligned}
RG(c, s) &= |(\omega_R \cdot R(c) - \omega_G \cdot G(c)) \ominus (\omega_R \cdot R(s) - \omega_G \cdot G(s))| \\
BY(c, s) &= |(\omega_B \cdot B(c) - \omega_Y \cdot Y(c)) \ominus (\omega_B \cdot B(s) - \omega_Y \cdot Y(s))| \\
O_\theta(c, s) &= \omega_\theta \cdot |O_\theta(c) \ominus O_\theta(s)| \\
\bar{C} &= \bigoplus_c \bigoplus_s N(RG(c, s)) - N(BY(c, s)) \\
\bar{O} &= \sum_\theta N(\bigoplus_c \bigoplus_s N(O_\theta(c, s))) \\
\bar{I} &= \bigoplus_c \bigoplus_s N(|I(c) \ominus I(s)|) \\
\bar{T} &= \bigoplus_c \bigoplus_s N(|T(c) \ominus T(s)|) \\
SM_{TD} &= \omega_I \bar{I} + \omega_C \bar{C} + \omega_O \bar{O} + \omega_T \bar{T}
\end{aligned}$$

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 would be 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:

$$e_{ROI}(\bar{\omega}) = \frac{\max(SM(\bar{\omega})) - \max(SM(\bar{\omega})|_{ROI})}{\max(SM(\bar{\omega}))}$$

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. 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 [65, 66].

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 [62, 75, 76]. 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{a} . 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.

When a set of optimized weights are found for a set of contexts (given a task), we train an artificial neural network [30] to associate between the context and the optimal weights. For more details on how this training is done we refer to our previous works [65, 66].

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 [31]. 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

$$\frac{dk}{dt} = -c \cdot k(t) + a \cdot \frac{E_{BU}(t)}{E_{TD}(t)}, \quad k = \begin{cases} 1 & k > 1 \\ k & 0 \leq k \leq 1 \\ 0 & k < 0 \end{cases}.$$

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 reach 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. If a single salient Top-down peak is attended to, 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.

4.4 Binocular cues for attention

Since the attentional system described above is generic with respect to the visual task, it might 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

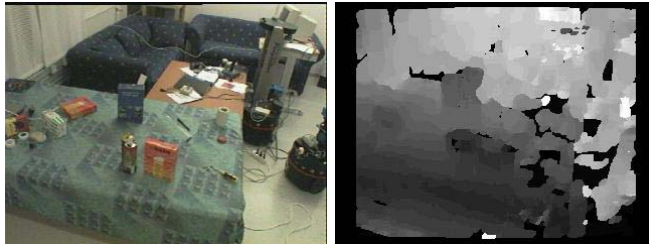


Fig. 6: Disparity map (right) of a typical indoor scene (left).

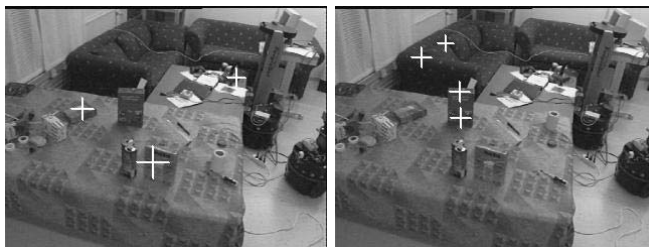


Fig. 7: Saliency peaks with saliency maps computed using top-down tuning for the orange package (left) and the blue box (right). The crosses reflect the sizes derived from the attentional process.

that in our robotic manipulator 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. 6, 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 manipulatable 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. 7, with crosses indicating the expected size of things in the scene.



Fig. 8: Disparity maps (right), prior foreground probabilities (center) and posteriori figure-ground segmentation (left).

5 Foveated segmentation

After an region of interest has been selected by the attentional system (see Section 4), 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 recognized and possibly manipulated. However, before this is done it would be beneficial if it could also be segmented from other 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 our system for foveated segmentation, the foreground probability of each pixel is computed in a probabilistic setting. From area based correlation we get a measurement for each pixel, measurements that are used to estimate the prior probability of a pixel belonging to the foreground. Examples of foreground priors can be seen in the center of Fig. 8.

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*. Fortunately, since there are only two possible labels the exact posteriori solution is given in a single graph-cut operation [25]. The resulting segmentation might look like the two images to the right in Fig. 8. These segmentations are then passed to either recognition or pose estimation. For more information on the precise modeling and motivations see [7].

5.1 From 3D segments to shape attributes

5.1.1 Without Table Plane Assumption.

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 [15]. 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. 10.

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.

5.1.2 With Table Plane Assumption.

We therefore complement our approach with a table plane assumption. Using a well-textured surface, it is possible to find the main plane and cut it with a 3D version of the Hough transform as in [32]. Following the table assumption the 3D points are mapped onto a 2D grid to easily find segments and basic shape attributes.

The result of transformation and clipping on the scene given in Fig. 9(a) can be seen in Fig. 9(b). The segmentation of objects is computed on the 2D grid (Fig. 9(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. 9(d))¹.

5.2 Associated attributes

The produced segments are just *things* [32], as the step to an *object* longs for semantics. One way to identify the semantics of a thing in order to derive an object 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), which mostly is measured in the quantitative domain. In our system, the basic intrinsic

¹ 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.

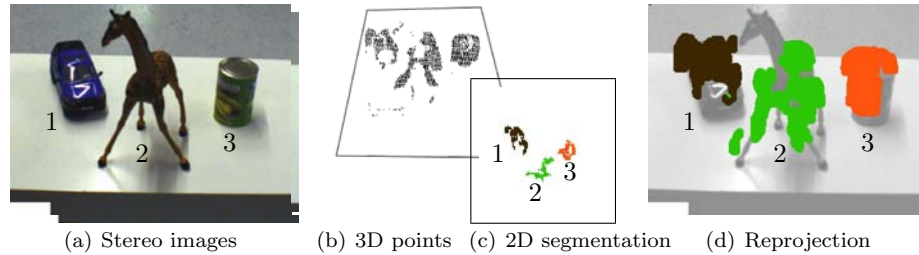


Fig. 9: Segmentation using the table plane assumption. Disparity information from the stereo images (a) produces 3D points (b). Having defined the dominant plane, the points can be projected onto this plane, where distinctive segments are computed (c) and reprojected to the image (d).

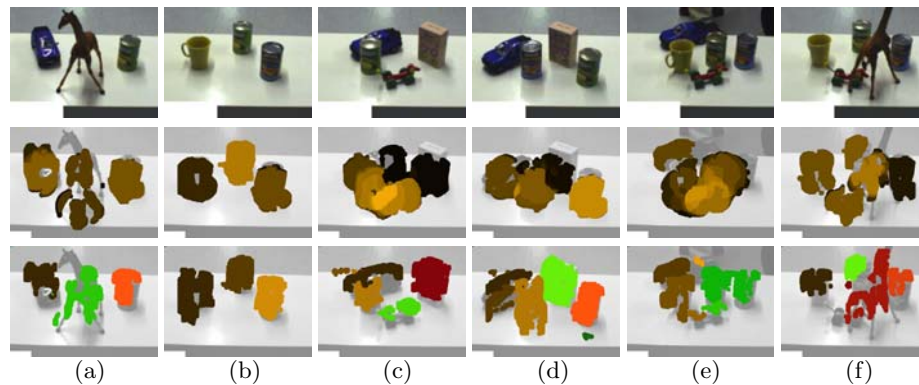


Fig. 10: Sample scenario segmentations (best viewed in color). Original images are shown in the first row. The second row shows results using the Mean Shift segmentation, the bottom row those using the table plane assumption (mentioned in Section 5.1.2). In the latter, (a) and (b) seem well segmented 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 unnoticed dog underneath the giraffe.

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 6.2) or shape approximation and grasping, as we proposed in [33].

6 Object Manipulation

To achieve real cognitive capabilities, robotic systems have to be able to interact with their environment. Thus, grasping and manipulating 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 [73]. Similar findings have been reported for the connectivity of other brain regions [53]. Recent neuroscientific findings state that predictions are a primary function of these connections, e.g., [64, 49]. The human brain, as the example of a truly cognitive system, uses predictions to focus attention and scale an otherwise too wide space of inputs.

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, [68, 10]. 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 on 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, [71, 2, 56, 14, 61, 8, 54]. 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 [51, 61, 41, 29, 13]. Another trend has focused on the use of machine learning approaches to determine the relevant features that indicate a successful grasp [50, 12, 37]. Finally, there have been efforts to use human demonstrations for learning grasp tasks [17].

One of the unsolved problems in robot grasping is the execution of grasps for novel objects in unstructured scenarios. For general settings, manipulation of unknown objects has almost not been pursued in the literature and it is commonly assumed that objects are easy to segment from the background. In the reminder of this section, we concentrate on an example of how the visual system presented so far can be used to provide grasping hypothesis for objects for which the identity/geometry is not known in advance. We acknowledge that



Fig. 12: Left) A left manipulation camera image, Middle) The corresponding disparity map, Right) Segmentation from mean shift in 3D space.

different machines.

6.2 Model-free manipulation

In general, we will not have a precise geometrical model for all objects the robot is supposed to manipulate. However, one can assume that the relation between the manipulation cameras and arm is approximately known. The manipulated objects are further known in terms of their intrinsic attribute lists and their state is complemented online with measurements of the extrinsic attributes. These lists (in particular the intrinsic one) are prepared off line and are part of the object model. For more details regarding these lists see [32].

6.2.1 Finding the orientation

In the current system, an object is commonly grasped along the normal of the plane on which they are. The reason for this is that the KUKA-arm is placed on a height of 50 cm (base height) and manipulating objects on a table height is restricted due to singularities. Even if the presented approach does not require the identity of the object to be known, it can be useful in order to make sure that the object is standing upright. Using the knowledge of the identity we can determine a suitable grasp incorporating the size and the shape of the object. 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. 12.

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

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

where the weighted mean position is $\bar{\mathbf{X}}_k$. Points away from the surface are suppressed through the weights

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

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 [41, 40].

7 Experimental Results

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)
- The dynamical combination of SM_{BU} and SM_{TD}

Correspondingly, the experiments presented below are divided 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. To the left in Fig. 13, a set of 10 objects used in the experiments are shown. These are all associated with a set of intrinsic attributes consisting of 3D size, appearance (SIFT and color based) and possible grasps given a certain orientation. The graph to the right 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. Note, also, that some cues are almost completely suppressed on some objects. Weight optimization was done for each object. The resulting set of triplets $\{ROI, \bar{\omega}_{opt}, \bar{\alpha}\}$ were used for training the neural networks (NN).

7.1.1 Weight optimization

It is important to understand that, even if one may reach a global minimum in the weight optimization (given the error function defined earlier), it does not necessarily mean that our Top-down map is “perfect” (like it is in Fig. 14). In fact, the Top-down map may not rank the correct ROI the highest, in spite of $e_{ROI}(\bar{\omega})$ being at its global minimum for that specific image and object. What this means is that for some objects $\min[e_{ROI}(\bar{\omega}_{opt})] \neq 0$, or simply that our

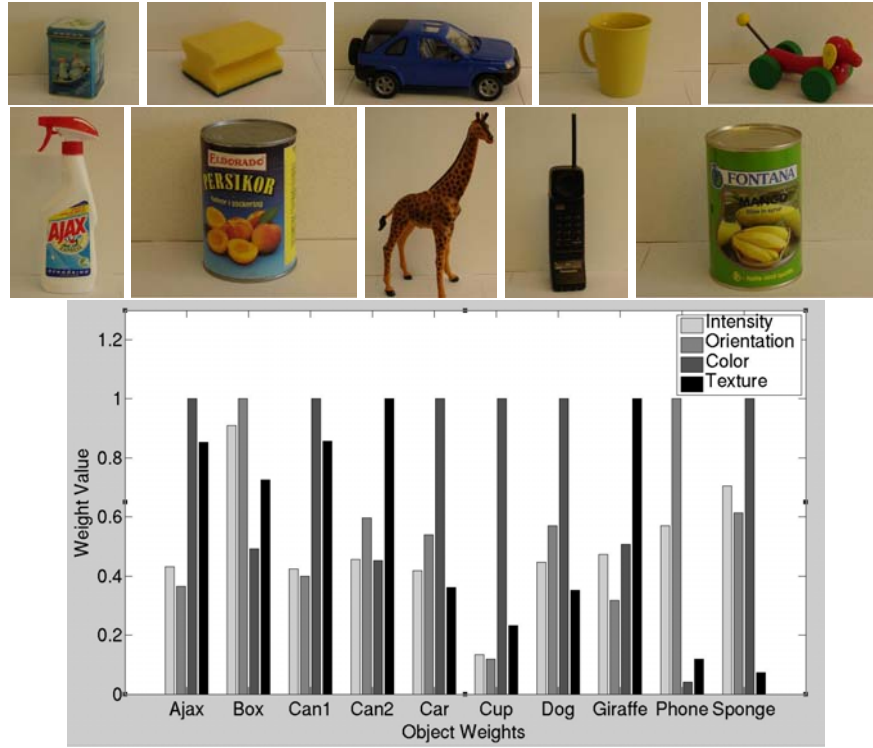


Fig. 13: A set of objects used for experiments (left) and the four TD-weights $\bar{\omega}_I, \bar{\omega}_O, \bar{\omega}_C, \bar{\omega}_T$ for each object in one particular image (right).

optimization method failed to find a proper set of weights the Top-down map at the desired location, as in the example Fig. 15.

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 ANN training

When performing the pattern association (equivalent with context learning) on the Neural Network it is important that the training data is “pure”. This means that only training data that gives the best desired result should be 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. 16. Results using



Fig. 14: 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).



Fig. 15: 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).

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 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 hypothesis validation

For object detection a large number of possible methods exist and they all have their individual characteristics. Unfortunately, there seem to be no single method suitable for all objects that might be of interest for a robotic system. Instead one has to rely on combinations of methods. Without providing an extensive study on all possible methods and combinations, we give an example that shows the benefit of foveated segmentation and multiple cues object recognition.

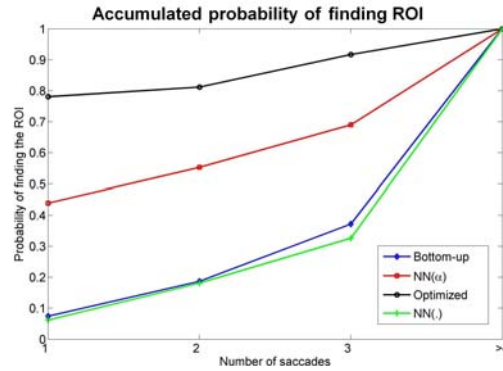


Fig. 16: The estimated accumulated probability of finding the ROI. The results were averaged over the entire test set of objects(ROI:s). BU is purely Bottom-up search, $NN_i(\bar{\alpha})$ is Top-down search guided by a Neural Network (trained on $i\%$ of the training data available) choosing context dependent weights, and $NN_i(\cdot)$ is the same without any context information.

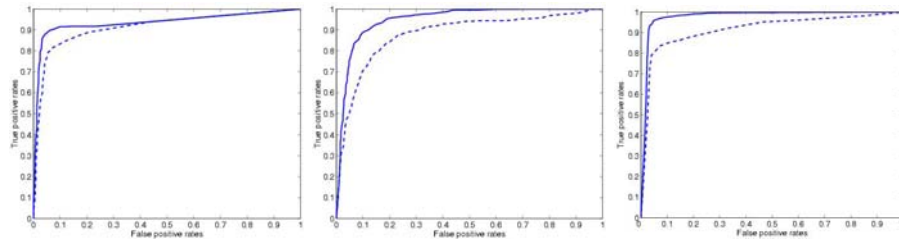


Fig. 17: ROC curves for SIFT based (left), color histogram based (middle) and combined (right) object detection, with (solid) and without (dashed) foveated segmentation.

For this purpose we have selected two methods that show different strengths and weaknesses. The first method is based on color histograms [24] and the other on scale and rotation invariant SIFT features [48]. Histogram based methods can be successfully used for uniformly colored objects without much texture, but tend to be sensitive to poor segmentations. Feature based method, on the other hand, work well even in cluttered environments, but break down when too few features can be extracted due to limited texture. In that sense the two methods are complementary.

We selected a larger set of 24 objects, similar to those in Fig. 13, and performed 886 object recognition tasks, using images provided in real-time using the binocular attention system described in earlier sections. The ROC curves in Fig. 17 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). For more details we refer to our previous work [7].

Since we are also interested in the manipulation aspect of the object recognition, we would want to combine the results of appearance and shape recognition². In other words we try to bind the object identity to its known intrinsic attributes. This binding serves two purposes: i) it will boost the recognition rate by disregarding more false positives, ii) it will allow 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 [32] as well as [9]. Since "Actions" in this paper 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) might be manipulated similarly.

7.3 Object Grasping - an example

Several experiments were performed on the robotic platform described in Section 6.1. The overall performance of the platform 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 specific task for the robot was to find certain objects and move them onto another table. This high-level task can be broken down into a couple of sub-tasks. The first sub-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. 18 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, preferably using the table-plane assumption mentioned in Section 5.1.2, was made on the "thing" of interest in that point. These segmentation results can be viewed as the enclosed regions in the foveal views of the four examples in Fig. 19. One consequence of real world conditions such as noise, varying illumination etc., is that these segmentations 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 (manipulations) on the object. Therefore, this imperfection is of minor importance, if the grasping yields a successful result.

If the segmented region turned out to contain the sought object in terms of appearance and other intrinsic attributes, the estimated position and orientation were sent to the manipulator. The system then chose an appropriate

² 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.

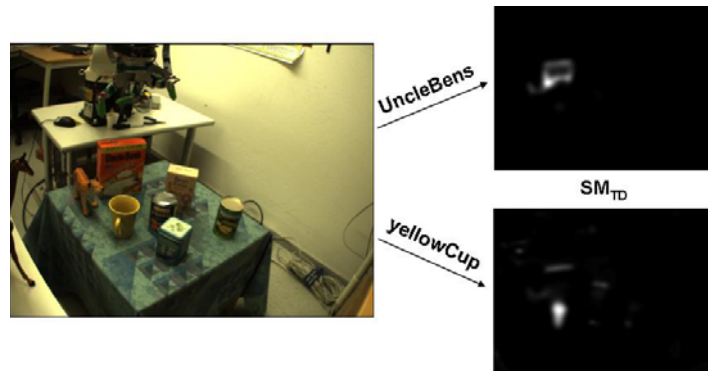


Fig. 18: Example with Top-down tuned saliency maps (UncleBens & yellowCup)



Fig. 19: The visual frontend. The top row shows the widefield 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.

manipulation (grasp) based on these intrinsic (identity) and extrinsic (state) attributes of the the object. In the final step the automatically chosen grasp was executed on the robotic platform, i.e. the robot arm drove to a location above the object so that the gripper was able to grasp the object.

A couple of examples of this complete chain are shown in Fig. 20. 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 works, where a more "natural" environment with clutter and distractors that might be of importance, will be encountered.

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 proven 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. 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.

In the development of a system like this, there is still a long way to go especially once the system is used for manipulation and robot control. Our current research concentrates on the evaluation and further development of the system in more complex manipulation scenarios.

Acknowledgments

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

References

- [1] C. Gaskett A. Ude and G. Cheng. Foveated vision systems with two cameras per eye. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3457–3462, Orlando, Florida, May 2006.
- [2] A. Bicchi and V. Kumar. Robotic grasping and contact: A review. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'00*, pages 348–353, 2000.
- [3] M. Björkman and J-O. Eklundh. Real-time epipolar geometry estimation of binocular stereo heads. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(3):425–432, 2002.
- [4] M. Björkman and D. Kragic. Combination of foveal and peripheral vision for object recognition and pose estimation. *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, 5:5135 – 5140, April 2004.
- [5] M. Björkman. *Real-Time Motion and Stereo Cues for Active Visual Observers*. Doctoral dissertation, Computational Vision and Active Perception Laboratory (CVAP), Royal Inst. of Technology, Stockholm, Sweden, 2002.
- [6] M. Björkman and J-O. Eklundh. Attending, foveating and recognizing objects in real world scenes. *Proceedings of British Machine Vision Conference, BMVC'04*, 2004.
- [7] M. Björkman and J.O. Eklundh. Foveated figure-ground segmentation and its role in recognition. In *Proc. British Machine Vision Conference*, volume II, pages 819–828, September 2005.
- [8] C. Borst, M. Fischer, S. Haidacher, H. Liu, and G Hirzinger. DLR hand II: Experiments and experiences with an antropomorphic hand. In *Proc. International Conference on Robotics and Automation*, volume 1, pages 702–707, September 2003.
- [9] R. Petrick N. Pugeault M. Steedman N. Krueger C. Geib, K. Mourao and F. Wörgötter. Object action complexes as an interface for planning and robot control. *IEEE RAS, Int Conf. Humanoid Robots(Genova):Dec. 4-6, 2006*, 2006.
- [10] U. Castiello. The neuroscience of grasping. *Nature Neuroscience*, 6:726–736, 2005.
- [11] S.B. Choi, S.W. Ban, and M. Lee. Biologically motivated visual attention system using bottom-up saliency map and top-down inhibition. *Neural Information Processing - Letters and Review*, 2, 2004.
- [12] J.A. Coelho, J. Piater, and R. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. *Robotics and Autonomus Systems*, 37:195–218, 2001.

-
- [13] J.A. Coelho Jr., J. Piater, and R. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. In *International Conference on Humanoid Robots*, Cambridge, Massachusetts, 2000.
- [14] J.A. Coelho Jr., K. Souccar, and R.A. Grupen. A control basis for haptically-guided grasping and manipulation. Technical Report CMP-SCI Technical Report 98-46, Dept. Computer Science, University of Massachusetts, Sept. 1998.
- [15] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [16] B. Draper and A. Lionelle. Evaluation of selective attention under similarity transforms. In *Proc. International Workshop on Attention and Performance in Computer Vision*, pages 31–38, 2003.
- [17] S. Ekvall and D. Kragic. Interactive grasp learning based on human demonstration. In *IEEE/RSJ International Conference on Robotics and Automation, ICRA '04*, 2004.
- [18] S. Ekvall and D. Kragic. Integrating object and grasp recognition for dynamic scene interpretation. In *IEEE International Conference on Advanced Robotics, 2005. ICAR '05*, pages 331–336, 2005.
- [19] S. Ekvall and D. Kragic. Receptive field cooccurrence histograms for object detection. In *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems, IROS'05*, pages 84–89, 2005.
- [20] S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Robotica*, 25:175–187, 2007.
- [21] S.M. Fairley, I.D. Reid, and D.W. Murray. Transfer of fixation using affine structure: Extending the analysis to stereo. *International Journal of Computer Vision*, 29(1):47–58, August 1998.
- [22] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [23] S. Frintrop. Vocus: A visual attention system for object detection and goal-directed search. *Lecture Notes in Computer Science*, 3899, 2006.
- [24] T. Gevers and A.W.M. Smeulders. Color based object recognition. *Pattern Recognition*, 32(3):453–464, 1999.
- [25] D.M. Grieg, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of Royal Statistical Society - B*, 51(2):271–279, 1989.

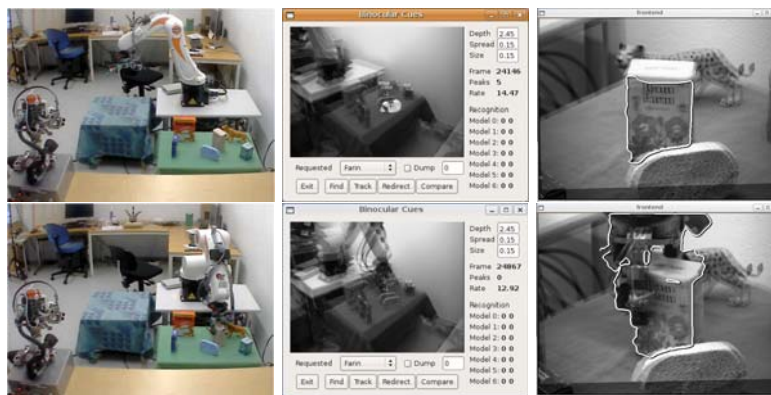
-
- [26] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the of the 4th Alvey Vision Conference*, 1988.
 - [27] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conference*, pages 147–151, 1988.
 - [28] R. Hartley and A. Zisserman, editors. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, 2000.
 - [29] A. Hauck, J. Rüttinger, M. Sorg, and G. Färber. Visual determination of 3D grasping points on unknown objects with a binocular camera system. In *Proc. International Conference on Robotics and Automation*, pages 272–278, Detroit, Michigan, May 1999.
 - [30] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ, 1994.
 - [31] Y. Hu, X. Xie, W-Y Ma, L-T. Chia, and D. Rajan. Salient region detection using weighted feature maps based on the human visual attention model. In *Proc. IEEE Pacific-Rim Conference on Multimedia*, pages 993–1000, 2004.
 - [32] K. Huebner, M. Bjorkman, B. Rasolzadeh, M. Schmidt, and D. Kragic. Integration of visual and shape attributes for object action complexes. 2008.
 - [33] K. Huebner, S. Ruthotto, and D. Kragic. Minimum Volume Bounding Box Decomposition for Shape Approximation in Robot Grasping. In *IEEE International Conference on Robotics and Automation*, 2008. To appear.
 - [34] L. Itti. *Models of Bottom-Up and Top-Down Visual Attention*. PhD thesis, California Institute of Technology, 2000.
 - [35] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, (2):194–203, 2001.
 - [36] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, November 1998.
 - [37] I. Kamon, T. Flash, and S. Edelman. Learning Visually Guided Grasping: A Test Case in Sensorimotor Learning. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):266–276, 1998.
 - [38] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, (4):219–227, 1985.
 - [39] T. Koike and J. Saiki. Stochastic guided search model for search asymmetries in visual search tasks. *Biologically Motivated Computer Vision*, pages 408–417, 2002.
 - [40] D. Kragic and Bjorkman. Strategies for object manipulation using foveal and peripheral vision. 2006.

-
- [41] D. Kragic, M. Bjorkman, H.I. Christensen, and J-O. Eklundh. Vision for robotic object manipulation in domestic settings. In *Proc. Robotics and Autonomous Systems*, volume 1, pages 85–100, July 2005.
- [42] D. Kragic and V. Kyrki. Initialization and system modeling in 3-d pose tracking. In *In IEEE International Conference on Pattern Recognition 2006. ICPR'06*, pages 643–646, Hong Kong, 2006.
- [43] Y. Kuniyoshi, N. Kita, K. Sugimoto, S. Nakamura, and T. Suehiro. A foveated wide angle lens for active vision. In *Proc. International Conference on Robotics and Automation (ICRA95)*, volume 3, pages 2982–2988, Nagoya, Aichi, Japan, 1995.
- [44] K. Lee, H. Buxton, and J. Feng. Selective attention for cueguided search using a spiking neural network. In *Proc. International Workshop on Attention and Performance in Computer Vision*, pages 55–62, Graz, Austria, July 2003.
- [45] Z. Li. A saliency map in primary visual cortex. *Trends in Cognitive Sciences*, 6(1):9–16, January 2002.
- [46] H. Longuet-Higgins. The interpretation of a moving retinal image. In *Philosophical Trans. Royal Society of London, B-208*, pages 385–397, 1980.
- [47] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, (293):133–135, 1981.
- [48] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Int'l Conf. Computer Vision (ICCV99)*, pages 1150–1157, Kerkyra, Greece, September 1999.
- [49] M. Matsumoto, K. Matsumoto, H. Abe, and K. Tanaka. Medial prefrontal cell activity signaling prediction errors of action values. *Nature Neuroscience*, 10:647–656, 2007.
- [50] A. Morales, E. Chinellato, A.H. Fagg, and A.P. del Pobil. Experimental prediction of the performance of grasps tasks from visual features. *International Journal of Humanoid Robotics*, 1(4):671–691, December 2004.
- [51] A. Morales, G. Recatalá, P.J. Sanz, and A.P. del Pobil. Heuristic vision-based computation of planar antipodal grasps on unknown objects. In *Proc. International Conference on Robotics and Automation*, pages 583–588, Seoul, Republic of Korea, May 2001.
- [52] J. Moren, A. Ude, A. Koene, and G. Cheng. Biologically-based top-down attention modulation for humanoid interactions. *International Journal of Humanoid Robotics*, 2008.
- [53] V. Mountcastle. An organizing principle for cerebral function: The unit model and the distributed system. In *The Mindful Brain (Gerald M. Edelman and Vernon B. Mountcastle, eds.)*. Cambridge, MA: MIT Press, 1978.

-
- [54] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko. Development of a high-speed multifingered hand system and its application to catching. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2666–2671, October 2003.
- [55] V. Navalpakkam and L. Itti. Sharing resources: Buy attention, get recognition. In *Proc. International Workshop Attention and Performance in Computer Vision*, Graz, Austria, July 2003.
- [56] A.M. Okamura, N.S. Smaby, and M.R. Cutkosky. An overview of dexterous manipulation. In *Proc. International Conference on Robotics and Automation*, pages 255–260, San Francisco, California, April 2000.
- [57] A. Oliva, A. Torralba, M.S. Castelhana, and J.M. Henderson. Top-down control of visual attention in object detection. In *Proc. International Conference on Image Processing*, pages 253–256, 2003.
- [58] B. Olshausen, C. Anderson, and D. van Essen. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, (13):4700–4719, 1993.
- [59] D. Paulus, U. Ahrlichs, B. Heigl, J. Denzler, J. Hornegger, M. Zobel, and H. Niemann. Active knowledge-based scene analysis. *Videre*, 1(4), 2000.
- [60] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. I. Christensen. Systems integration for real-world manipulation tasks. In *IEEE International Conference on Robotics and Automation, ICRA’02*, volume 3, pages 2500 – 2505, 2002.
- [61] R. Platt Jr., A.H. Fagg, and R. Gruppen. Nullspace composition of control laws for grasping. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1717–1723, Lausanne, Switzerland, 2002.
- [62] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *International Conference on Computer Vision*, 2007.
- [63] O. Ramström and H.I. Christensen. Object detection using background context. In *Proc. International Conference of Pattern Recognition*, pages 45–48, 2004.
- [64] R. Rao and D. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999.
- [65] B. Rasolzadeh. Interaction of bottom-up and top-down influences for attention in an active vision system. Master’s thesis, Royal Institute of Technology, Stockholm, Sweden, 2006. TRITA-CSC-E 2006:117, ISSN-1653-5715.

-
- [66] B. Rasolzadeh, M. Björkman, and J.O. Eklundh. An attentional system combining top-down and bottom-up influences. In *Proc. International Cognitive Vision Workshop (ICVW06)*, 2006.
- [67] Babak Rasolzadeh, Alireza Tavakoli Targhi, and Jan-Olof Eklundh. An attentional system combining top-down and bottom-up influences. In *WAPCV*, pages 123–140, 2007.
- [68] M.W. Riddoch, G.W. Humphreys, S. Edwards, T. Baker, and K. Wilson. Seeing the action: neuropsychological evidence for action-based effects on object selection. In *Nature Neuroscience*, 4, pages 84–88, 2001.
- [69] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14(3):365–372, 1980.
- [70] B. Scassellati. A binocular, foveated, active vision system. Technical report, MIT AI Memo 1628, March 1998.
- [71] K.B. Shimoga. Robot grasp synthesis: A survey. *Proc. International Journal of Robotics Research*, 3(15):230–266, June 1996.
- [72] Y.C. Shiu and S. Ahmad. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $ax = xb$. 5:16–29, 1989.
- [73] N. Sigala and N.K. Logothetis. Visual categorization shapes feature selectivity in the primate temporal cortex. *nature*, 415:318–320, 2002.
- [74] A. Sloman. Evolvable biologically plausible visual architectures. In *British Machine Vision Conference, BMVC'01*, pages 313–322, 2001.
- [75] T.M. Strat and M.A. Fischler. Context-based vision: Recognition of natural scenes. pages 532–536, 1989.
- [76] T.M. Strat and M.A. Fischler. The use of context in vision. 1995.
- [77] P. Azad J. Schroder A. Bierbaum N. Vahrenkamp R. Dillmann T. Asfour, K. Regenstein. Armar-iii: An integrated humanoid platform for sensory-motor control. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 169–175, 2006.
- [78] A. Tavakoli Targhi, E. Hayman, J.O. Eklundh, and M. Shahshahani. The eigen-transform and applications. In *Proc. Asian Conference on Computer Vision*, pages 70–79.
- [79] E. A. Topp, D. Kragic, P. Jensfelt, and H. I Christensen. An interactive interface for service robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 3469–3475, New Orleans, April 2004.

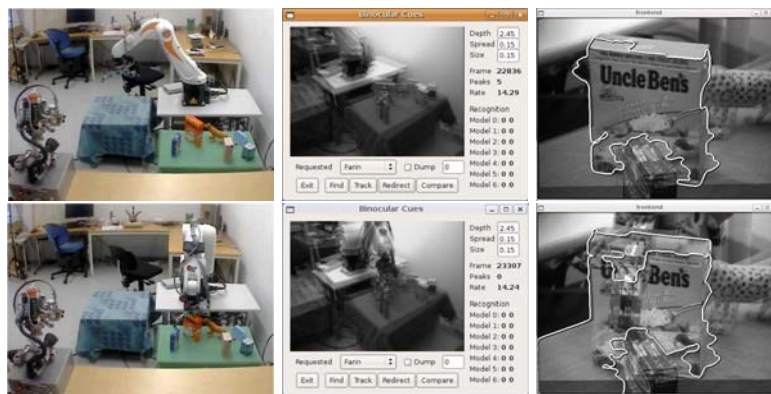
-
- [80] A.M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.
 - [81] R.Y. Tsai and R.K. Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. 1988.
 - [82] J.K. Tsotsos. Analyzing vision at the complexity level: Constraints on an architecture, an explanation for visual search performance, and computational justification for attentive processes. 1987.
 - [83] J.K. Tsotsos. A 'complexity level' analysis of immediate vision. 1(4):303–320, January 1988.
 - [84] J.K. Tsotsos. The complexity of perceptual search tasks. 1989.
 - [85] Y. Ye and J.K. Tsotsos. Sensor planning in 3D object search. *Computer Vision and Image Understanding*, 73(2):145–168, 1999.



(Farin)



(Tiger)



(UncleBen)

Fig. 20: 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)

Active Multi-View Object Search on a Humanoid Head

Kai Welke, Tamim Asfour and Rüdiger Dillmann

University of Karlsruhe (TH), IAIM, Institute of Computer Science and Engineering (CSE)
P.O. Box 6980, 76128 Karlsruhe, Germany

{welke,asfour,dillmann}@ira.uka.de

Abstract—Visual search is a common daily human activity, which is a prerequisite to the interaction with objects encountered in the environment. Humanoid robots supposed to take part in human daily life should possess similar capabilities in terms of representing, recalling and attending to objects of interest.

In this paper, we introduce the so-called Feature Ego-Sphere (FES) as scene memory for a multi-view object representation in a humanoid robot and define associated processes necessary for the identification of object locations in the scene. The experimental results have been carried out on an active humanoid head equipped with both, perspective and foveal stereo camera systems. The perspective view is used to generate hypotheses of object locations, which are verified by directing the gaze of the foveal cameras towards potential regions of interest.

I. INTRODUCTION

The object search process is a common daily human activity. Almost all actions that humans perform rely on specific items which support the action e.g. as tools. For example, drinking requires a cup, eating a fork, and writing requires a pencil. While such an object search task is natural to humans it is still hard to implement on a technical system.

In the context of human visual perception, the pop-out effect is a well known phenomenon which supports the guidance of attention towards a specific object within a cluttered scene. According to [1] and [2], the pop-out is attributed to the interplay between dorsal and ventral pathways of the human visual system and is modelled using a blackboard architecture, which strongly relies on parallel processing and distributed representations of objects.

For technical systems, the visual search task has often been formulated as top-down attention guidance. Different approaches in the literature modulate the output of a bottom-up attention system using e.g. the feature-gate technique [3]. While these approaches follow the line of biological plausible systems it is hard to achieve good results for arbitrary objects due to the parallel and distributed nature of the problem.

In this work, we propose an approach which takes into account the difference between the "wetware" used for processing in human brains and the hardware of technical systems. Instead of successively filtering the visual stimuli starting with low-level cues as in traditional attention systems [4], our approach starts with a search for the object in the scene with coarse features. Using the resulting matches, we follow a hypothesis and test procedure in order to verify the matches with local, more descriptive features. With this ap-

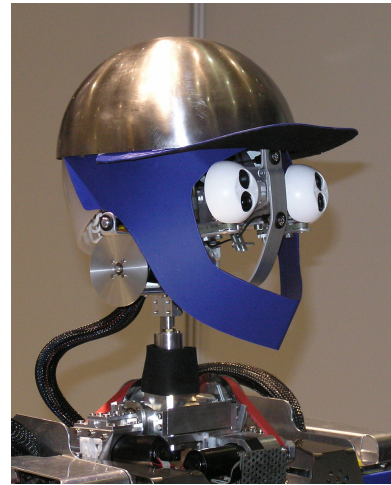


Fig. 1. The Karlsruhe Head is equipped with a 3DoF active camera system and offers one perspective and one foveal camera pair.

proach it is possible to decompose the required search space and reduce the computational complexity of the problem.

The goal of our work is to equip a humanoid robot with the ability to identify known objects, which have previously been acquired by the robot itself. The approach of acquisition on the system has a multitude of advantages. First, online acquisition allows the robot to explore its environment in order to incrementally acquire world knowledge. Second, the approach produces representations within the sensor space of the system. This makes them easy to apply in perceptual processes. While such representations usually combine information from different senses (e.g. haptics, auditory system, vision), we currently only consider visual information, since it can be acquired on the system [5]. Objects in the system are represented using a multi-view appearance-based description. This allows the localization of objects from all viewpoints, as necessary in natural tasks. In the current state 3D shape information is not considered, since the acquisition of shape is mainly achieved using the haptic system and is still hard to be acquired online.

In order to store the object information from the current scene as collected during the object search process, a scene memory is required. In the following we propose a scene memory which assures the persistence and consistence of already acquired information about the scene. As will be shown, the scene memory allows for the integration of

multiple hypotheses based on spatial coherence, which makes the search task more robust.

The target platform for our experiments is the Karlsruhe Humanoid Head [6]. As shown in Fig. 1, the head provides two pairs of active stereo cameras. One pair with wide angle lenses for perspective views and one pair using a small angle of view, which allows a more detailed visual inspection and resembles the fovea of the human visual system. The proposed system uses both camera systems to actively analyze the current scene. Hypotheses of object locations are extracted in the perspective view using coarse descriptors of the object’s appearance. Based on the hypotheses, eye movements are executed to direct the gaze of the foveal cameras to the corresponding location and to verify the hypotheses using more detailed features.

With the availability of the necessary technical systems, a large number of capable vision systems for humanoid robots has been presented the last years. The proposed approach stands in a line with systems that exploit the use of foveated vision. In [7], the authors present a vision system which integrates foveal and perspective information on a humanoid robot. Object detection is performed in the perspective image. Once a known object is detected, the gaze of the foveal camera is directed towards the object for recognition using a PCA-based approach. Once the object is recognized, the robot points its hand towards the object. The system proposed in [8] can be considered as state-of-the-art in this field. The authors make use of the perspective cameras to calculate hypothetical locations in the scene for a given object using its 3D size and hue cues. The gaze of the foveal camera is directed towards the hypotheses in order to perform recognition using SIFT features. Furthermore, segmentation on basis of disparity maps is performed. The system works in real-time and takes into account multiple canonical views of objects. The authors also perform experiments which motivate the use of foveal cameras for the recognition task.

The approach proposed in this paper makes also use of the foveal images in order to gain the ability of a more detailed analysis of hypotheses locations in a scene, as describe in the above work. Unlike the systems described above, our approach constructs a consistent and persistent scene memory during the visual search task, that is constantly verified and which can be used for successive visual tasks.

II. ACTIVE MULTI-VIEW OBJECT SEARCH

A. System Overview

Fig. 2 illustrates the memories and processes involved in the object search task. The input of the system consists of the foveal and perspective camera image pairs as provided by the Karlsruhe Humanoid Head and the ID of an object to search for. The search process generates hypotheses for locations which correspond to the provided object ID and updates the scene memory accordingly. The attention process serializes the verification process by guiding the gaze of the foveal camera pair to salient locations in the scene. Each new gaze initiates a new process to verify the hypotheses in the scene memory using the more detailed images from the

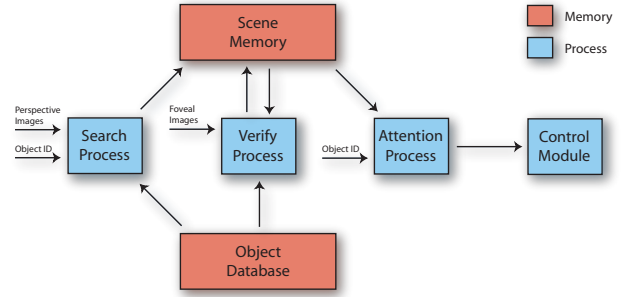


Fig. 2. Overview of the memories and processes involved in the proposed system. The search process generates hypotheses of object locations. From these hypotheses, the attention and control processes generate a gaze sequence for the foveal camera pair which is used for subsequent verification steps. The object database contains appearance based multi-view representations of acquired objects. The system iteratively assures consistent and persistent information in the scene memory.

foveal camera pair. The goal of the procedure is to update the scene memory with consistent and persistent locations of the searched object in the scene. The information from the scene memory can then be used for further visual or interaction tasks.

In the following, we describe the different parts of the system depicted in Fig. 2.

B. Object Database

The object database contains all object specific information which is required during the object search procedure. As described in section I object search is performed only on the basis of multiple views of objects, which facilitates the acquisition on the humanoid robot and thus in the sensor space defined by the robotic system.

In the current state of the work, the object views are generated off-line in the object modelling center [9]. The object modelling center offers a camera pair mounted on a robotic arm which controls the zenith of the current view and a turn table which controls the azimuth. The target positions for recording object views are calculated by subdividing an icosahedron in two stages, which results in 66 equally distributed camera positions around the object. Due to the limits of the robotic arm, the zenith only covers views between $\pm 75^\circ$.

The collected views are then stored in an aspect graph representation ([10],[11]). In our system, the aspect graph is modelled as bidirectional spherical graph which ts of one node per stored view, where the edges between nodes are generated using Delaunay triangulation and thus express the neighbourhood of views. The aspect graph serves as basis for the feature extraction process. For each view, one global and one local descriptor is extracted and associated to the corresponding node.

In the current implementation of the system we make use of color cooccurrence histograms (CCH) [12] as global descriptors. CCHs offer a description of the object, which is invariant to the rotation in the viewing plane and robust

towards scaling. Furthermore, they combine texture information (in terms of information about pairs of neighbored pixels) as well as color information. We currently use histograms which cover the hue channel of the HSV image. As local descriptors we use the scale invariant feature transform approach (SIFT)[13]. Each SIFT descriptor is stored together with a reference vector to the origin of the image.

In order to reduce the size of required memory, features are clustered into similar groups using the BIRCH [14] clustering approach for feature quantization. For this work, we compared the performance of the BIRCH algorithm with the Growing Neural Gas (GNG) method which we used in earlier work [15] and its incremental version IGNG. We observed that the BIRCH algorithm produces similar clustering results as the IGNG with superior efficiency. Both algorithms support incremental clustering, which is required to allow the incremental acquisition of object representations. In contrast to the GNG and IGNG, where the number of generated clusters depends on the maximum accumulated error per cluster (see [15]), the BIRCH algorithm produces a clustering of the feature space which fits into a given amount of memory.

After feature quantization all cluster centroids are stored in a feature pool. Furthermore, for each object, a feature graph is generated which has identical structure as the aspect graph. The nodes of the feature graph contain references to the corresponding clusters in the feature pool. The object database then consists of one feature graph per object and one feature pool.

The feature pool itself is implemented as a two-level hierarchical memory. All features are held on disk, while the memory only contains a limited amount of features. Features are cached in memory during instantiation and removed from memory following the least recently used (LRU) strategy.

C. Scene Memory

A visual scene memory is necessary to provide a consistent visual model of the observed scene. It has been shown that human perception accumulates such a scene memory "across separate glances and over time" [16]. In our work, the scene memory contains information about matches between searched objects and the current scene associated with spatial information. These matches are successively verified by moving the foveal cameras to salient locations in the scene. Together with the processes specified in sections II-D, II-E and II-F, the scene memory provides consistent information about objects and their locations accumulated over time. The information is constantly verified and is persistently made available for visual tasks.

The scene memory proposed in this work is constructed as an ego-sphere. The application of ego-spheres as sensory memory is usually called Sensory Ego-Sphere (SES). In [17], the authors introduce the SES as sphere around the so called ego-center which is typically located in the base coordinate frame of the robot. The entries in the SES correspond to sensory stimuli and are stored with $2\frac{1}{2}D$ information using their spherical polar coordinates azimuth (ϕ), zenith (θ) and

their distance from the ego-center (r), thus forming an ego-centric representation of the current scene. The SES has been used in a number of different applications such as multi-modal bottom-up attention [18] and image mapping and visual attention [19].

In contrast to the SES, where usually sensory information is stored, we introduce the Feature Ego-Sphere (FES) as scene memory. The FES has the structure of an ego-sphere, however, instead of sensory stimuli as typically stored in the SES, information about matches between features from the object database and the current scene are stored as nodes. Thus, the FES contains the knowledge gathered so far by comparing stored object representations with the current scene. Particularly, the FES does not only contain information about positive matches, but also retains information about the falsification of hypotheses.

Despite its function as scene memory, the FES supports the proposed hypothesis and verify approach in different ways. First it allows the integration of different hypotheses on a basis of spatial coherence. Neighboured entries in the FES which describe the same object can be combined to a common node and thus increase the certainty of the corresponding match. Second, the FES can be deployed to generate the necessary attentional shifts required to verify the hypotheses (see section II-F).

The FES memory structure offers two different types of nodes which are motivated by the hypothesis and verify approach for object search:

- *hypothesis node*: In addition to the position $p_h = (\phi, \theta, r)$, information about the match between searched features and the object is stored. The hypotheses node can contain pointers to verify nodes. Hypotheses nodes are made persistent in the scene memory in order to allow the search module to detect changes in the scene.
- *verify node*: A verify node results from the verification of a node (either verify or hypothesis node). It contains the verified position p_v and information about the match between verified features and the scene.

The content of the FES is manipulated by two basic operations:

- *addEntry*: Adds a hypothesis node to the FES. The node is only added if there is no similar hypothesis node already present in the proximity. If there is a hypothesis node present which contains different data, change is detected and the hypothesis node is adapted.
- *verifyEntry*: This operation is called once a node of the FES has been verified. If the verified node is a hypothesis node, a new verify node is created and linked to the hypothesis node. If the verified node is a verify node, its position and match is updated. If the adaptation of the position moves the verify node in proximity of another verify node, both nodes are combined if they contain similar data.

Hypothesis nodes are generated by the search process (see section II-D) using the perspective view of the cameras. The gaze is directed towards salient hypotheses and the verify

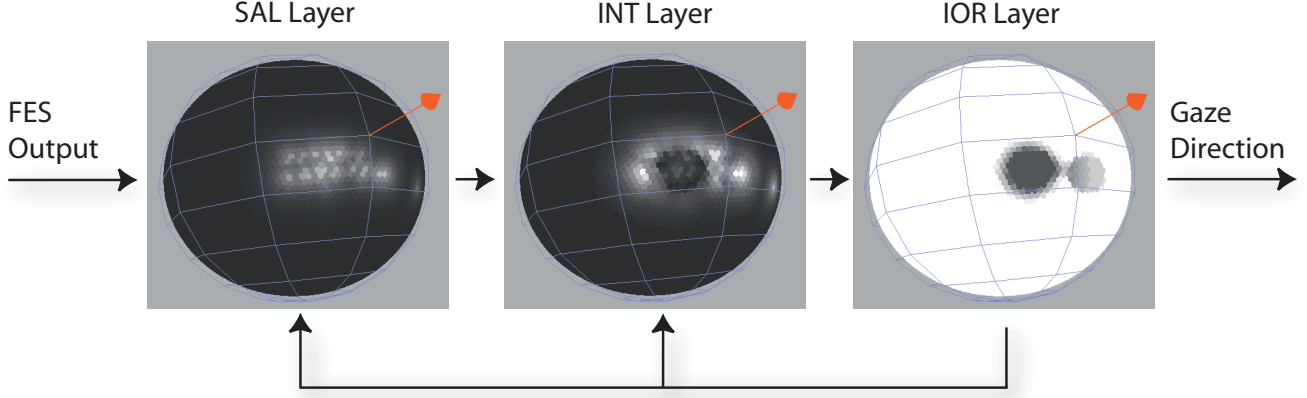


Fig. 3. The spherical WTA network as used to direct the gaze of the robot head. The network consists of the saliency layer (SAL), integration layer (INT) and inhibition-of-return layer (IOR). The input saliency is generated from nodes of the FES that correspond to the attended object. Leaky integrate-and-fire neurons in the INT layer initiate an attentional shift once a threshold of activation is reached.

process invalidates or verifies the corresponding nodes using the foveal camera views. The verification process successively generates a verify node, if not already present, with corrected position and links it to the hypothesis. Multiple verify nodes are combined to one node if they represent the same object and similar positions. In the course of the verification process, verify nodes are moved towards valid object positions in the scene.

D. Search Process

The search process is responsible for the generation of possible positions accompanied with the quality of the match given a specific object. As input, the perspective image pair from the robots camera system and the object ID to search for are required. In order to determine hypotheses about object position, the search process requests all CCH clusters as stored in the feature graph of the corresponding object. The search is accomplished using an integral images approach. Object positions are accepted on base of the histogram intersection with the database features. 3D positions are generated using the disparity map calculated on the perspective image pair.

The resulting hypotheses comprise the position of the hypotheses and the result of the histogram intersection as quality of the match. Using the *addEntry* operation of the FES (see section II-C) a new hypothesis node is added to the scene memory if not already present.

E. Attention Process

The attention process determines the sequence for the verification of the FES content. Two factors influence the decision which FES nodes to verified next: the quality of the corresponding match and the elapsed time since the last verification. Such problems of selective attention can be solved using a winner-take-all (WTA) network as introduced in [4].

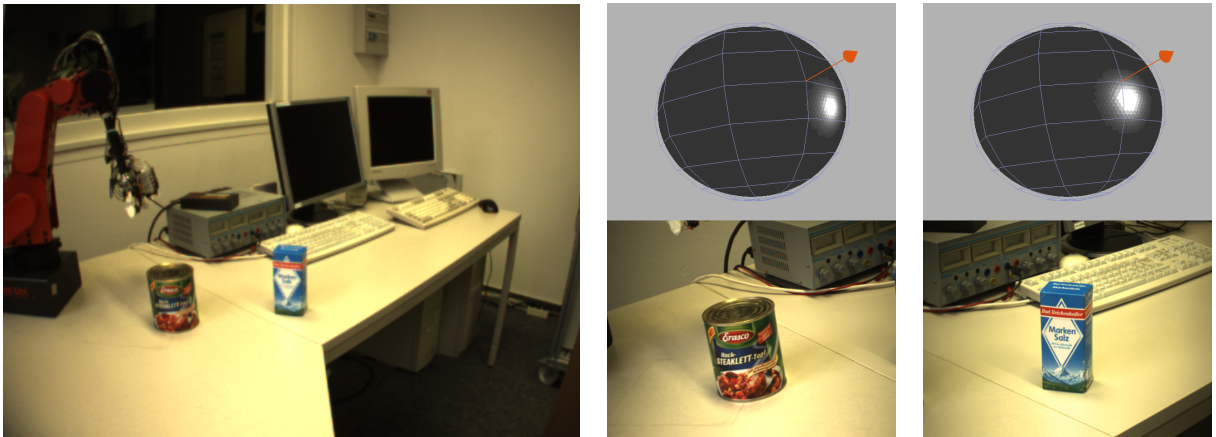
In order to provide the necessary input for the WTA network, the FES content is filtered using the object ID of the currently attended object. From the content of the FES, a saliency sphere is generated. The saliency sphere represents a traditional saliency maps on the ego-sphere. Each leaf node from the FES, which corresponds to the attended object ID, generates a stimulus represented as 2D gaussian with amplitude proportional to the stored match quality on the saliency sphere. Multiple stimuli are combined using a MAX operator.

The saliency sphere is then used as input for a spherical implementation of the WTA network. Fig. 3 shows the three layers of the network. The saliency sphere is modulated with the feedback from the inhibition of return (IOR) layer in the SAL layer. The resulting activations are integrated using leaky integrate-and-fire neurons in the INT layer. Once the activation of a node in the INT layer exceeds a threshold, the neuron fires and generates activation in the IOR layer with 2D gaussian shape. The inverse output of the IOR layer is used as feedback to the SAL and INT layers.

Each time a neuron in the INT layer fires, a new saccadic eye movement is initiated in order to direct the gaze of the foveal camera system towards the corresponding FES entries. For this purpose, all FES nodes which have similar spherical polar coordinates (θ, ϕ) are determined and the gaze is directed towards the closest node (using the third component r).

F. Verify Process

The verify process is responsible for the constant verification of the FES content. Each time the gaze of the robot is adapted by the attention process, a new verification cycle is initiated. As input, the detailed foveal views corresponding to the current gaze are available. The verification process requests all nodes from the FES, which are visible within the current gaze. Using the match and object ID from the



(a) Example scene setup used for the object search experiments viewed from the left perspective camera. Two objects were presented to the system.

(b) Resulting saliency sphere and foveal view for the soup can search task.

(c) Resulting saliency sphere and foveal view for the salt box search task.

Fig. 4. Results of the object search experiments for two objects.

hypothesis node, the SIFT features for all associated object views are determined using the feature graph.

The object's presence is verified by filtering the SIFT matches using a 2D Hough space and voting for the center of the object (see [20]). The corresponding match is thresholded and used to modulate the quality previously associated with the node.

In order to adapt the position of the object encoded in the node, the distance of the object locations and the center of the foveal image of left and right foveal camera is determined. Since no stereo calibration is available for the foveal cameras, the current target position for the inverse kinematics of the perspective cameras (see II-G) is adapted to move the foveal cameras closer to the object's position. Note that using this approach the FES finally contains positions which point the gaze of the foveal cameras towards the object.

For each verified node the operation *verifyEntry* of the FES is called which updates the content of a verify node or generates a new one.

G. Head Control Module

The head control module is responsible for the generation of the posture of the head-eye system. There are essentially two possible strategies to execute the required movements: closed-loop control and open-loop control. In closed-loop control, usually visual feedback is used in order to derive the position error of the eyes iteratively. In contrast open-loop control does not depend on visual feedback but uses the kinematic model of the system to determine the desired posture. Since the target posture in the context of our work is defined by the spike of a single neuron in the WTA network, the necessary visual feedback for closed-loop control cannot be provided.

In order to control the head using the open-loop strategy, the kinematic model of the head-eye system has to be determined. Therefore a kinematic calibration process is performed. We use the approach introduced in [21], which yields accurate results since it avoids methodical errors

which are usually introduced with the assumption of two intersecting rotation axes.

The inverse kinematic problem is solved on the basis of the calibrated kinematic model. Since only eye movements are used in the system, the problem can be formulated as non-redundant mapping from 3d Cartesian space to 3D joint angle space (for more details see [6]). We use the inverse Jacobian approach to solve for the joint angles of the perspective camera system. Furthermore, the stereo calibration of the perspective camera system is available in order to provide the disparity map required for the search process.

III. EXPERIMENTAL RESULTS

A. Setup

For the experiments presented in this section, five objects were stored in the object database. The object view acquisition generated 58 views per object covering equidistant angles in the range of $\theta = [-75^\circ; 75^\circ]$ and $\phi = [0^\circ; 360^\circ]$. The resulting 290 CCH descriptors used in the search module were quantized to 75 cluster centers. In average, each object view generated about 1700 SIFT descriptors. For the quantization of SIFT features, 200000 cluster centers were extracted for the verification process.

The Karlsruhe Humanoid Head was equipped with a pair of $4mm$ lenses for the perspective cameras and $12mm$ lenses for the foveal cameras. For the experiments the objects were positioned in about $1m$ distance of the head.

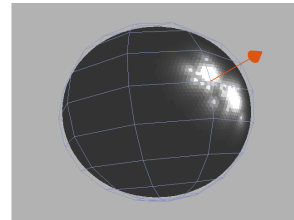
B. Experiment I: Object Search Task

The task in the first experiment consisted in a simple object search. Two objects out of the acquired object set were presented in front of a non-uniform background. An example input scene is depicted in Fig. 4(a).

In separate search tasks each of the two objects from the input scene was enquired. Fig. 4(b) shows the results of the search for the soup can. The upper image illustrates the saliency sphere after 22 verifications. All incorrect hypotheses have been eliminated and the correct hypotheses have



(a) Scene setup used for the complex search task. Two instances of one object are presented to the system in a cluttered scene.



(b) Resulting saliency sphere and foveal views of the left foveal camera. In the final state, the system focusses alternatingly on the position of both object instances.

Fig. 5. Results of the object search task for two instances in a complex scene.

been fused by the FES to form a combined position estimate on the saliency sphere. The lower picture in Fig. 4(b) shows the foveal image of the left camera. Similar results could be achieved for the salt box (see Fig. 4(c)). The search task for the salt box could be completed within 9 verification steps.

The number of required verification steps depends on the number of hypotheses generated by the search process. Considering the input scene in Fig. 4(a), the salt box has an outstanding color signature and could be brought into focus on the first saccade. The remaining 8 verifications adapted the positions of the verify nodes to a single estimate in the FES. In contrast, the CCH of the soup can was found in multiple incorrect image parts as e.g. the robot arm and the teach box. These spurious hypotheses could be invalidated by performing additional saccadic eye movements and verification steps.

The same procedure was performed for all objects in the object database. Similar results could be achieved with a mean number of required verifications steps of 17 in order to retrieve a saliency sphere similar to Fig. 4(b) and Fig. 4(c).

C. Experiment II: Complex Search Task

The goal of the second experiment was to perform a more complex scenario. The cluttered scene shown in Fig. 5(a) which contains a large amount of distractor objects was presented to the system. The task of this experiment consisted in finding both instances of the cereal box among the distractor objects.

The system performed a saccade containing 28 verification steps in order to retrieve the results depicted in Fig. 5(b). Two locations of high intensity are visible on the saliency sphere, which correspond to the positions of both object instances. Despite the two peaks, other local intensity maxima are visible. These result from unverified hypotheses which lie in the proximity of highly activated areas. If the activation of such hypotheses is below a threshold they can be dominated by the strong stimuli nearby. In order to also remove these

local maximas, the IOR size can be reduced which results in a prolonged verification procedure. After the 28 iterations, the gaze alternately focussed the two instances of the cereal box.

The scene memory generated during the search task is depicted in Fig. 6. From the initially large amount of hypotheses nodes only a small amount could be verified and has been associated to verify nodes. All other hypotheses nodes are either invalidated or dominated by a stronger stimulus in the proximity. The system produced one unique verify node per instance of the object in the scene.

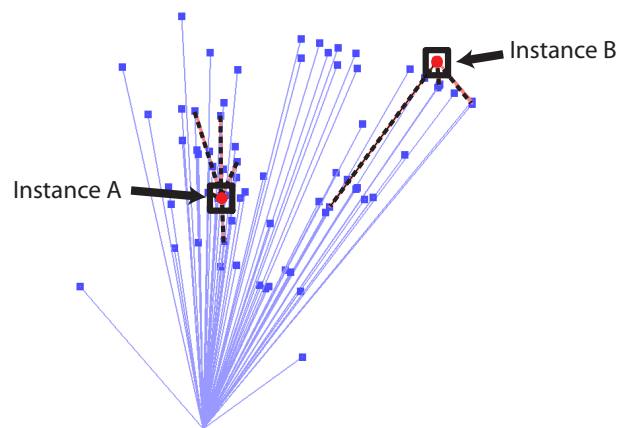


Fig. 6. Content of the FES after 28 eye movements and verifications. Both instances of the searched object have one unique verify node (marked by the black box), which is supported by multiple hypotheses nodes (associations marked with dotted line). For each hypotheses node the connection to the ego-center is drawn.

D. Discussion

For the experiments we abstain from giving recognition rates of the system since the object search performance directly depends on the object to find in the search task.

Because the approach relies on CCH and SIFT features, it is limited to objects and object views which exhibit properties that allow a robust representation with the considered descriptors. The CCH implementation using the hue channel cannot handle object views that contain black and white in major parts. This results from the fact that black and white do not have a unique representation in HSV color space. Furthermore, lighting is an issue when using color descriptors. The experiments were carried out using natural lighting conditions with variations during the day. Reducing the threshold for CCH matching allowed to account for small changes in ambient lighting, but increased the number of invalid hypotheses produced during the search process. Since the verification process is not that critical concerning lighting, good results could still be achieved.

To provide good verification performance using SIFT features, enough texture has to be present on the object in order to provide the necessary number of corner points. Whereas logos and picture printed on the objects provide good features, small written text is usually not covered by the SIFT descriptor. In our database the short side views of the objects usually contained text and large white areas and thus could not be consistently identified. For other views, e.g. as presented in the previous sections, the results could be reproduced consistently.

The proposed system currently runs on a single core 3.0 GHz linux PC. Each verification step took about 20 seconds. We did not use an optimized implementation of the feature matching process. The approach is intended and already prepared to run on our vision cluster which comprises 6 IBM eServer connected via Ethernet. We expect that by means of optimization and cluster implementation we will reach a verification run-time of less than 1 second.

IV. CONCLUSIONS

In this work we presented an approach which provides persistent and consistent information about object locations resulting from an object search task. The FES datastructure and associated processes were introduced as scene memory, the necessary processes and modules required to perform a visual object search task were presented and discussed. Experiments comprising the search for one object at a time and the search for multiple instances of an object in cluttered scenes were carried out and discussed.

The results show that even for complex tasks the proposed hypothesis and verify approach is able to identify the object locations by actively analyzing the scene.

V. ACKNOWLEDGMENTS

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

REFERENCES

- [1] F. van der Velde and M. de Kamps, "A model of visual working memory in PFC," *Neurocomputing*, vol. 52-54, pp. 419–424, 2003.
- [2] F. van der Velde, M. de Kamps, and G. T. van der Voort van der Kleij, "CLAM: Closed-loop attention model for visual search," *Neurocomputing*, vol. 58-60, pp. 607–612, 2004.
- [3] M. Wright, J. Chodzko, and D. Luk, *Biologically Motivated Computer Vision*. Springer Berlin / Heidelberg, 2000, ch. Development of a Biologically Inspired Real-Time Visual Attention System, pp. 779–785.
- [4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [5] D. Omrcen, A. Ude, K. Welke, T. Asfour, and R. Dillmann, "Sensorimotor processes for learning object representations," in *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2007.
- [6] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann, "The Karlsruhe Humanoid Head," in *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2008, (submitted to).
- [7] A. Ude, C. G. Atkeson, and G. Cheng, "Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2003, pp. 2173–2178.
- [8] M. Björkman and J.-O. Eklundh, "Vision in the real world: Finding, attending and recognizing objects," *International Journal of Imaging Systems and Technology*, vol. 16, pp. 189–208, 2003.
- [9] R. Becher, P. Steinhaus, R. Zöllner, and R. Dillmann, "Design and implementation of an interactive object modelling system," in *Proc. Conference Robotik/ISR 2006*, 2006.
- [10] J. Koenderink and A. van Doorn, "The singularities of the visual mapping," *Biological Cybernetics*, vol. 24, no. 1, pp. 51–59, 1976.
- [11] —, "The internal representation of solid shape with respect to vision," *Biological Cybernetics*, vol. 32, pp. 211–216, 1979.
- [12] P. Chang and J. Krumm, "Object recognition with color cooccurrence histogram," in *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.
- [13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, p. 11501157.
- [14] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM-SIGMOD International Conference on Management of Data*, 1996, pp. 103–114.
- [15] K. Welke, E. Oztop, G. Cheng, and R. Dillmann, "Exploiting similarities for robot perception," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3237–3242.
- [16] D. Melcher, "Persistence of visual memory for scenes," *Nature*, vol. 26, 2001.
- [17] R. A. Peters II, K. E. Hambuchen, K. Kawamura, and D. M. Wilkes, "The sensory ego-sphere as a short-term memory for humanoids," in *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2001.
- [18] J. Ruesch, M. Lopes, A. Bernardino, J. Hornstein, J. Santos-Victor, and R. Pfeifer, "Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 962–967.
- [19] K. A. Fleming, R. A. Peters, and R. E. Bodenheimer, "Image mapping and visual attention on a sensory ego-sphere," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 241–246.
- [20] P. Azad, T. Asfour, and R. Dillmann, "Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, 2007.
- [21] M. Przybylski, K. Welke, T. Asfour, and R. Dillmann, "Kinematic calibration of an active camera system," in *Proc. IEEE/RAS International Conference on Humanoid Robots (HUMANOIDS)*, 2008, (submitted to).